DTU Library

Communication Technologies Support to Railway Infrastructure and Operations

Sniady, Aleksander

Link to article, DOI:
10.11581/DTU:00000010

Publication date:
2015

Document Version
Publisher's PDF, also known as Version of record

Communication Technologies Support to Railway Infrastructure and Operations

Aleksander Sniady

Ph.D. Thesis
May 2015

DTU Fotonik
Department of Photonics Engineering

Communication Technologies Support to Railway Infrastructure and Operations

Aleksander Śniady

Ph.D. Thesis

Networks Technology & Service Platforms
DTU Fotonik
Technical University of Denmark

May 2015

To my parents and grandparents.

Supervisors:
José Soler
Lars Dittmann

Technical University of Denmark
DTU Fotonik
Department of Photonics Engineering
Ørsteds Plads, Building 343,
2800 Kongens Lyngby, Denmark
www.fotonik.dtu.dk

RobustRailS

This thesis is a part of RobustRailS project, which is funded by *The Danish Council for Strategic Research*.
www.robustrails.man.dtu.dk

DOI: 10.11581/DTU:00000010

Cover photo credits: Rafał Żmuda

Abstract

GSM-Railways (GSM-R), which is state-of-the-art railway mobile communication technology, is gradually replacing legacy analogue radio systems. Although GSM-R is an unquestionable achievement in terms of European railway interoperability, from a telecommunication point of view, it is an obsolete technology.

In the research work presented in this thesis, GSM-R technology is analysed and its main shortcomings are identified, namely: lack of capacity, limited data transmission capabilities, and inefficiency in radio resource usage. Due to these significant disadvantages, alternative mobile technologies are considered to replace GSM-R in the future.

This thesis is focused on Long Term Evolution (LTE) as one of the most likely successors to GSM-R. As a technology designed for commercial purposes, LTE has to be investigated specifically in railway environment. Using computer-based simulations, the LTE network is examined in various scenarios modelling typical railway conditions. The transmission performance offered by LTE is analysed under worst-case assumptions in terms of traffic load, base station density, and user speed. The results demonstrate that LTE fulfils transmission requirements set for the two most important railway applications: European Train Control System (ETCS) signalling and railway-specific voice communication. Therefore, LTE is technically capable of replacing GSM-R as the communication network for the European Rail Traffic Management System (ERTMS).

Moreover, the simulation results show that LTE offers a significant improvement over GSM-R in terms of transmission capacity and performance. Thus, LTE as a

railway communication technology would create an opportunity to introduce new business-supporting applications, which could enhance railway operation. The demand for such applications is growing in railways, but the GSM-R networks cannot deliver them.

Furthermore, a radio access architecture based on cooperating macro and micro cells is proposed in the thesis. This heterogeneous network architecture, which is novel for railways, may bring numerous advantages, such as high network availability and reduction of inter-cell handover rate for running trains. It also enables railways to use new high-frequency radio bands, which is not a feasible option in the classical railway radio deployments. Simulation results indicate that the macro/micro architecture offers huge capacity increase, which can be used for providing bandwidth-demanding applications, such as video surveillance.

All in all, this thesis presents a feasible evolution in the field of railway communications. LTE technology together with the novel heterogeneous architecture may transform railway mobile networks from being a bottleneck of the system into becoming its strong asset.

Résumé

Radiokommunikationssystemet GSM-R (GSM-Railways), der er det nyeste system til jernbaneformål, er ved at erstatte ældre analoge radiosystemer. Selv om GSM-R er et ubetinget fremskridt hvad angår de europæiske jernbaners samarbejde, så er det teknologisk set en forældet teknologi.

Forskningsarbejdet, der præsenteres i denne afhandling, analyserer GSM-R teknologien og identificerer de vigtigste ulemper: mangel på kapacitet, begrænset mulighed for data transmission og dårlig udnyttelse af radio ressourcerne. På grund af disse betydelige ulemper ser vi på alternative mobilteknologier, der kan erstatte GSM-R i fremtiden.

Denne afhandling fokuserer på systemet LTE (Long Term Evolution), som er en af de mest sandsynlige afløsere for GSM-R. Da LTE-teknologien er udviklet til generelle kommercielle formål, må den analyseres med henblik på specifik anvendelse i jernbane-miljø. Ved hjælp af computerbaserede simuleringer undersøges LTE-netværk under værst tænkelige scenarier med hensyn til trafikbelastning, tæthed af basisstationer og brugerhastighed. Resultaterne viser, at LTE opfylder de krav, der stilles til de to vigtigste jernbaneanvendelser: signalering i European Train Control System (ECTS) og jernbane-specifik talekommunikation. LTE er derfor i stand til at erstatte GSM-R som kommunikationsnet for European Rail Traffic Management System (ERTMS).

Simulationsresultaterne viser endvidere, at LTE vil være et betydeligt fremskridt i forhold til GSM-R med hensyn til transmissionskapacitet og ydeevne. LTE vil

således som kommunikationsteknologi åbne op for introduktion af nye forretningsunderstøttende anvendelser, som vil kunne styrke jernbanedriften. Efterspørgslen efter sådanne anvendelser er voksende, men GSM-R netværk kan ikke tilbyde dem.

I afhandlingen foreslås der endvidere en arkitektur for radionetadgang baseret på samspil mellem makro- og mikroceller. Denne heterogene netværksarkitektur, der er ny inden for jernbane kommunikationsnet, tilbyder talrige fordele så som høj tilgængelighed og reduceret hyppighed for handover for kørende tog. Den tillader også jernbanerne at bruge nye højfrekvensradiobånd, hvilket ikke er en mulighed i klassiske jernbane kommunikationsnet. Simuleringsresultaterne viser, at makro/mikro arkitekturen åbner op for en enorm kapacitetsforøgelse, som gør det muligt at tilbyde båndbreddekrævende anvendelser, så som videoovervågning.

Sammenfattende anviser denne afhandling en mulig udvikling inden for jernbanekommunikation. LTE-teknologien i samspil med en ny heterogen arkitektur gør det muligt at transformere mobile netværk fra at være en flaskehals i jernbanesystemer til at være et stærkt aktiv.

Acknowledgements

I would not be able to complete the work that is presented in this thesis without help of many people who were around me during my Ph.D. studies.

First and foremost, I would like to thank my supervisors: José Soler and Lars Dittmann, whose guidance was behind everything that I have achieved.

I am grateful to Marion Berbineau and Mohamed Kassab for their hospitality and fruitful collaboration. Also, thanks to the participants of RobustRailS project for all of the interesting meetings and knowledge sharing.

I would like to thank the members of Networks Technology & Service Platforms group, who created such a nice working environment. Special thanks to Ying Yan, Michael Berger, Villy Bæk Iversen, Cosmin Caba, Jahanzeb Farooq, Vaidas Karosas, and Szymon Śniady for proof-reading and helpful comments on this thesis.

Thanks to my family and friends. I cannot imagine completing these studies without encouragement from my parents and support of my friends—those here in Denmark, those a bit further away and those at the other side of the world.

The most special thanks to Joana, whose invaluable help, support, and love accompanied me during the good and the bad days.

I would also like to express my gratitude to The Danish Council for Strategic Research—the primary founder of my Ph.D. work—and also to Tourbo Fonden, Otto Mønsteds Fond, and Institut Français in Denmark for the grants that allowed me to conduct my research and to publish the results.

A. Sniady

Aleksander Śniady

Contents

List of Figures

List of Tables

List of Acronyms

3GPP	Third Generation Partnership Project
AM	Acknowledged Mode
AMR	Adaptive Multi-Rate
ARP	Allocation and Retention Priority
ATO	Automatic Train Operation
ATP	Automatic Train Protection
BLER	BLock Error Rate
BSC	Base Station Controller
BSR	Buffer Status Report
BSS	Base Station Subsystem
BTM	Balise Transmission Module
BTS	Base Transceiver Station
CA	Carrier Aggregation
CBTC	Communication Based Train Control
CoMP	Coordinated Multi-Point
CP	Cyclic Prefix
CSCF	Call Session Control Function
CSD	Circuit-Switched Data
CSFB	Circuit-Switched Fall Back
DMI	Driver Machine Interface

DNS	Domain Name System
E-CSCF	Emergency CSCF
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
EC	European Commission
EIRENE	European Integrated Radio Enhanced NEtwork
eMLPP	Enhanced Multi-Level Precedence and Pre-emption
eNodeB	E-UTRAN NodeB
EoMA	End of Movement Authority
EPC	Evolved Packet Core
EPS	Evolved Packet System
ERA	European Railway Agency
ERTMS	European Rail Traffic Management System
ETCS	European Train Control System
EU	European Union
EVC	European Vital Computer
FA	Functional Addressing
FDD	Frequency-Division Duplex
GBR	Guaranteed Bit Rate
GMSK	Gaussian Minimum Shift Keying
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communication
GSM-R	GSM-Railways
GSMA	Global System for Mobile Association
GTP	GPRS Tunnelling Protocol
HARQ	Hybrid Automatic Retransmission Request
HLR	Home Location Register
HSPA	High Speed Packet Access
HSS	Home Subscriber Register
I-CSCF	Interrogating CSCF
ICI	Inter-Carrier Interference
ICIC	Inter-Cell Interference Coordination
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPsec	Internet Protocol Security
ISDN	Integrated Services Digital Network
ISI	Inter-Symbol Interference
ITS	Intelligent Transportation Systems
LDA	Location Dependent Addressing

LTE	Long Term Evolution
LTE-A	LTE-Advanced
MA	Movement Authority
MAC	Medium Access Control
MCS	Modulation and Coding Scheme
MIMO	Multiple Input Multiple Output
MME	Mobility Management Entity
MORANE	Mobile Oriented RAdio NEtwork
MOS	Mean Opinion Score
MPTCP	Multipath TCP
MS	Mobile Station
MSC	Mobile Switching Center
NSS	Network and Switching Subsystem
OBU	On-board Unit
OFDM	Orthogonal Frequency-Division Multiplexing
OFDMA	Orthogonal Frequency-Division Multiple Access
OTT	Over The Top
P-CSCF	Proxy CSCF
P-GW	Packet Data Network Gateway
PCRF	Policy and Charging Rules Function
PDCCH	Physical Downlink Control CHannel
PDCP	Packet Data Convergence Protocol
PDSCH	Physical Downlink Shared CHannel
PHY	Physical Layer
PoC	Push-to-talk over Cellular
PRACH	Physical Random Access CHannel
PSAP	Public Safety Answering Point
PUCCH	Physical Uplink Control CHannel
PUSCH	Physical Uplink Shared CHannel
QAM	Quadrature Amplitude Modulation
QCI	QoS Class Identifier
QoS	Quality of Service
QPSK	Quadrature Phase-Shift Keying
RBC	Radio Block Centre
RBG	Radio Bearer Group
REC	Railway Emergency Call
RLC	Radio Link Control
RRC	Radio Resource Control

RSAP	Railway Safety Answering Point
RTP	Real-time Transport Protocol
S-CSCF	Serving CSCF
S-GW	Serving Gateway
SAE	System Architecture Evolution
SC-FDMA	Single-carrier FDMA
SDP	Session Description Protocol
SG	Scheduling Grant
SINR	Signal-to-Interference-and-Noise Ratio
SIP	Session Initiation Protocol
SR	Scheduling Request
SV-LTE	Simultaneous Voice and LTE
TCC	Traffic Control Centre
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TETRA	Terrestrial Trunked Radio
TTT	Time-To-Trigger
UDP	User Datagram Protocol
UE	User Equipment
UHF	Ultra High Frequency
UIC	International Union of Railways
UM	Unacknowledged Mode
UMTS	Universal Mobile Telecommunication System
UNISIG	Union Industry of Signalling
UTO	Unattended Train Operation
VBS	Voice Broadcast Service
VGCS	Voice Group Call Service
VLR	Visitor Location Register
VoIP	Voice over IP
VoLGA	Voice over LTE via Generic Access
VoLTE	Voice over LTE
WiMAX	Worldwide Interoperability for Microwave Access
ZC	Zadoff-Chu

Ph.D. Publications

[Sniady2012a] A. Sniady, J. Soler, and L. Dittmann. “Communication Technologies Support to Railway Operations”. In *Danish Railway Conference 2012*, Copenhagen, Denmark, May 2012. Poster presentation.

[Sniady2012b] A. Sniady and J. Soler. “An overview of GSM-R technology and its shortcomings”. In *Proceedings of the 12th International Conference on ITS Telecommunications (ITST)*, pages 626–629, Taipei, Taiwan, November 2012. IEEE. ISBN 978-1-4673-3071-8.

[Sniady2013a] A. Sniady and J. Soler. “Performance of LTE in High Speed Railway Scenarios”. In *Proceedings of the 5th International Workshop on Communication Technologies for Vehicles, Nets4Cars / Nets4Trains 2013*, volume 7865 of *Lecture Notes in Computer Science*, pages 211–222, Villeneuve d’Ascq, France, May 2013. Springer. ISBN 978-3-642-37973-4.

[Sniady2013b] A. Sniady, J. Soler, and L. Dittmann. “Future alternatives to GSM-R”. In *Danish Railway Conference 2013*, Copenhagen, Denmark, May 2013. Poster presentation.

[Sniady2013c] A. Sniady and J. Soler. “Evaluation of ETCS performance with LTE as alternative railway communication network using OPNET”. In *Proceedings of OPNETWORK 2013*, Washington, D.C., USA, August 2013. Distinguished Technical Paper Award.

[Sniady2013d] A. Sniady and J. Soler. "Impact of the traffic load on performance of an alternative LTE railway communication network". In *Proceedings of the 13th International Conference on ITS Telecommunications (ITST 2013)*, pages 396–401, Tampere, Finland, November 2013. IEEE.

[Sniady2013e] A. Sniady and J. Soler. "Can LTE become an alternative to GSM-R?". In *Strategisk forskning i transport og infrastruktur konference*, Kgs. Lyngby, Denmark, June 2013. Abstract presentation.

[Sniady2014a] A. Sniady, M. Kassab, J. Soler, and M. Berbineau. "LTE Micro-cell Deployment for High-Density Railway Areas". In *Proceedings of the 6th International Workshop, Nets4Cars / Nets4Trains / Nets4Aircraft 2014*, volume 8435 of *Lecture Notes in Computer Science*, pages 143–155, Offenburg, Germany, May 2014. Springer. ISBN 978-3-319-06643-1.

[Sniady2014b] A. Sniady and J. Soler. "LTE for Railways: Impact on Performance of ETCS Railway Signaling". In *IEEE Vehicular Technology Magazine*, volume 9(2), pages 69–77. IEEE, June 2014.

[Sniady2014c] A. Sniady and J. Soler. "Capacity gain with an alternative LTE railway communication network". In *Proceedings of the 7th International Workshop on Communication Technologies for Vehicles, Nets4Cars-2014-Fall*, pages 54–58, St. Petersburg, Russia, October 2014. IEEE.

[Sniady2015a] A. Sniady, M. Sønderskov, and J. Soler. "VoLTE Performance in Railway Scenarios". In *Proceedings of the 2015 Joint Rail Conference (JRC2015)*, number JRC2015-5723, San Jose, CA, USA, March 2015. ASME.

[Sniady2015b] A. Sniady, J. Soler, M. Kassab, and M. Berbineau. "Ensuring ETCS data integrity in LTE". In *IEEE Vehicular Technology Magazine*, 2015. Under review at the moment of submission of this thesis.

CHAPTER 1

Introduction

"To the working of railways, the telegraph had become essential."
Robert Stephenson, 1856

Communication technologies have always played a crucial role in railway systems. No other mode of transport is more directly dependent on its communication technology than railways. This is due to the basic nature of a railway system, which is characterized by two features. The first one is a distributed infrastructure—consisting of many interrelated movable components—which must be controlled and supervised. The second feature is the long braking distance of a rail vehicle, which often exceeds the driver visibility distance [1, pp. 17–18]. Together, these two features create a railway system: many slow-braking trains sharing a complicated dynamic infrastructure. Operation of this complex system requires reliable and timely information about the train movements and the state of the infrastructure elements. Throughout the railway history, this exchange of information has been provided by various communication technologies.

Capabilities and performance of the railway communication systems affect railway operations. For instance, having faster information flow, train dispatching decisions can be made faster. By increasing communication reliability, the probability of travel delays due to communication failures is reduced. The more precise and detailed information available, the higher safety can be guaranteed. Therefore, railways have often been early adopters of new communication solutions and technologies that offered benefits in terms of efficiency, safety, and capabilities.

1.1 Brief history of the railway communication

The close interdependency between communications and railways is visible on the example of the electrical telegraph. The Great Western Railway in England was one of the first places where the telegraph was successfully implemented for commercial usage [2]. This first trial connection started operation in 1838, but already in 1856 the telegraph was called an "*indispensable companion of railways*". By that time, only in Great Britain, 7200 miles of telegraph links were deployed and over one million of messages was transmitted annually [3]. Telegraph was used mainly to inform whether line interconnecting stations was clear or blocked. However, it also offered other "services". The most interesting was the ingenious method for informing about an accident. A train driver, in case of an accident or other serious problem, was cutting the telegraph wire. The link breakdown was then detected by a station officer, who was aware that something has happened on the line. Hence, the telegraph greatly contributed to the railway safety.

Since the telegraph age, railways have adopted various technologies. Some of these technologies enabled communication between dispatchers at neighbouring train stations, e.g. telephony. Other technologies allowed for communication between dispatchers and train drivers, i.e. train signalling. Throughout its history, railways used hand signals, ball signals, flags, telegraph, semaphore, and position lights, until this evolution brought wireless radio and colour light signals used today [1, pp. 179–183]. Each new generation enriched the communication and enabled exchange of more detailed information. Thanks to this, new services and procedures could be introduced to improve safety and efficiency of the rail transport.

Nowadays, railways move towards digital wired and wireless networks, such as Ethernet and GSM-Railways (GSM-R). These networks, besides the traditional person-to-person communication, also offer connectivity between computer-based systems. For instance, they allow real-time message exchange between a computer-based Traffic Control Centre (TCC) and a locomotive computer unit. The information delivered in this way is more detailed and more precise than it could ever be possible with the colour light signals. Besides, the risk of a human error is significantly reduced.

Digital communication between trains and the TCC enabled developing of command-control systems that support and supervise train drivers. These systems include features such as in-cab signalling, which provides detailed information about the speed limits and the distance to the End of Movement Authority (EoMA).

Such precise information could not be delivered via colour light signals. Besides, this information is now displayed right on the driver desk, so there is no risk that a driver will miss a trackside signal. Another example of a feature supporting drivers is the Automatic Train Protection (ATP). ATP supervises train movements and ensures that the train stops before its EoMA, i.e. it prevents the train from passing a “stop” signal [1, pp. 208–211]. Neither in-cab signalling nor ATP could be possible without a real-time communication between the train and the TCC.

The European Train Control System (ETCS) [4] and the Communication Based Train Control (CBTC) [5, 6] are the best examples of the communication-based command-control systems. ETCS and CBTC provide not only in-cab signalling and ATP, but they also offer moving block operation, reduction in trackside equipment, emergency communication, and many more. Some of the CBTC systems go as far as eliminating human drivers entirely and providing the so-called Unattended Train Operation (UTO) [7].

Without systems like ETCS, high-speed railways would not be possible. Similarly, without CBTC, efficient high-frequency metro railways could not be built. Both ETCS and CBTC operate on the basis of the underlying mobile communication technologies. Therefore, for modern railways, mobile network is as essential as the telegraph was for the 19$^{\text{th}}$ century railways.

1.2 Future evolution

In-cab signalling and ATP are breakthrough features greatly improving railway safety and efficiency. However, the demand for communication-based services does not end here. Especially seeing the rapid developments in wireless telecommunication technologies in recent years, more advanced railway communication-based services can be envisioned. Various research publications and technical reports propose exemplary services for the future railways and their passengers:

- Real-time video streaming allowing train drivers to monitor level-crossings or other points of potential danger,
- Real-time information about movement of other trains [8],
- Video surveillance monitoring train interior for passenger safety [8],
- Remote access to data generated by train on-board sensors, which would improve monitoring, diagnosis, and preventive maintenance [9, pp. 38–39],
- Remote access to documentation for on-board and trackside staff (e.g. during maintenance tasks),
- Cargo tracking and a management system combined with on-board sensors,

- Passenger information and entertainment services [9, 10],
- Electronic ticketing systems [10],
- Internet access for passengers [9, p. 22],
- Services supporting multi-modal transportation,
- Remote safety monitoring via smoke detectors, platform supervision, etc. [7].

In the future, some of these services may be widely used, while others could be abandoned. It is also very likely that new unforeseen ideas will emerge. Regardless of which services exactly will be developed and used by railways, it is widely accepted that the popularity and importance of communication-based services is going to grow [11, p. 44]. New applications will contribute to safer, more efficient and more reliable railways.

The current mobile communication standards used by the mainline railways in Europe are not able to provide these new services. This is because of the limited capacity and poor data transmission capabilities offered by these standards. Therefore, as the demand for the communication-based services will grow, railways will have to deploy newer networks with more capable technologies.

Together with the new services, railway dependability on the communication technologies will increase even more. Capabilities and reliability of the communication technology chosen by railways will determine the capabilities and reliability of the railways themselves. Therefore, a good and reliable communication will be a basis for good and reliable railway system. On the other hand, limitations and failures of the communication systems will become limitations and failures of the whole railways. Thus, choice of the communication network for railways is of crucial importance.

1.3 Structure of the thesis

This thesis gathers and summarizes research outcomes of the Ph.D. project on the future railway communication technologies. Railways require various types of communication: on-board communication, train-to-ground communication, communication between interlocking and trackside objects, communication between dispatchers and many more. This Ph.D. work is concerned only with the train-to-ground communication systems.

The thesis is based on research work that has been published in several journal and conference papers [Sniady2012a] – [Sniady2015b].

The rest of the thesis is organized as follows:

- **Chapter 2** describes the state-of-the-art in European railway mobile communication, namely the GSM-R network. The main shortcomings of GSM-R are identified and analysed. These shortcomings are used as the starting point for a discussion on the future railway communication network. In the recent years, various mobile technologies have been considered as possible successors of GSM-R. The most notable of these technologies are presented and briefly evaluated.

- **Chapter 3** investigates Long Term Evolution (LTE) as a possible future mobile communication technology for railways. One of the main purposes of railway mobile network is to provide ETCS signalling. Therefore, ETCS transmission performance over LTE is investigated in a series of simulation scenarios. The performance offered by LTE is confronted with ETCS requirements.

- **Chapter 4** considers railway voice communication over LTE. Despite the growing importance of digital signalling, voice communication remains a crucial application for railways. Voice over LTE (VoLTE) standard is presented. Its performance is validated in railway scenarios and confronted with the requirements set by the railway industry.

- **Chapter 5** proposes a novel heterogeneous macro/micro radio access architecture for railways. Its purpose is to increase radio capacity, improve network availability and optimize cell deployment. Furthermore, mechanisms for protecting ETCS data integrity in dense LTE deployments are investigated.

- **Chapter 6** concludes the thesis and presents the outlook on the future of railway mobile communication.

Additionally, there are two appendices attached to this thesis. They present technical details of the simulation work:

- **Appendix A** includes source code of the simulation models.

- **Appendix B** presents all details on the simulation scenarios.

CHAPTER 2

Motivation

Railways are large and complex systems. They are built, expanded and upgraded gradually over the years. While new technologies are deployed on new and upgraded rail lines, the old solutions remain in use elsewhere. Therefore, the lifetime of railway technologies is usually counted in decades. This technological diversity of railways is increased by the country-specific standards. Traditionally, each country developed its own systems more or less independently from the neighbours. Due to these reasons, railway technological landscape usually consists of a variety of incompatible systems and standards. This applies to electrification system, track gauges, platform height and many more. However, this diversity is most visible in railway communication and signalling technologies.

This technological diversity is a major obstacle in international train operation. The problem is especially visible in Europe, where over 20 train command-control system are in use [12, 13]. These systems often have very similar purpose and functionality. However, since they are incompatible, cross-border train operation is unnecessarily complicated. Locomotives must be equipped with multiple command-control and communication systems. This brings economical and operational disadvantages. For instance, these additional command-control systems consume the limited space in the locomotive. They also increase the locomotive price (control-command systems may be as much as 25% of the price [12, p. 25]). Besides, drivers must be qualified to use each of them.

At the same time, in Europe, the demand for international train travels is very high and growing. It is not uncommon for a train to cross multiple country borders. This is why European countries realized already at the end of 1980s that an inter-operable command-control system must be developed. This is how the work on the European Rail Traffic Management System (ERTMS) started.

Chapter organization

The following sections of this chapter present the ERTMS and its two elements: European Train Control System (ETCS) and GSM-Railways (GSM-R). Since this thesis is focused on train-to-ground communication systems, GSM-R is presented in more detail, including its major shortcomings. These shortcomings are the motivation for research on alternative communication technologies for railways. The most important candidates for the future GSM-R replacement are presented.

2.1 European Rail Traffic Management System

The European Rail Traffic Management System (ERTMS) is the first international standard for train command-control and train-to-ground communication. In the late 1980s, European countries realized that segmentation of the railway market becomes a significant problem for the future development of rail transport [12, p. 31]. The lack of interoperability was especially problematic for high-speed railways, whose advantages could not be fully realized without cross-border services. Therefore, the European railway industry began work on a common standard.

Development of ERTMS involved a broad representation of the railway industry and European institutions: European Union (EU) bodies, International Union of Railways (UIC), railway operators gathered in ERTMS User Group, Union Industry of Signalling (UNISIG), and—from 2004—the European Railway Agency (ERA). This process started in 1989, when the first studies and research was initiated by the European Commission (EC) [1, p. 240]. Development and tests continued for many years and resulted in the initial specifications, which were published in April 2000 [12, p. 74]. However, the work on ERTMS has been continued and many updated specifications have been published since then.

The most important features and advantages brought by ERTMS are as follows [4, 12]:

- Improvement of railway interoperability by establishment of new common European standards, which allows uninterrupted cross-border train operation.
- Introduction of a new command-control system for high-speed trains, which would eventually replace legacy command-control standards in Europe.

- Increase of efficiency and safety of high-speed trains due to in-cab signalling and Automatic Train Protection (ATP).
- Increase of the track capacity by usage of the moving block concept and dynamic braking curves.
- Reduced complexity of train driver work, thanks to a single standardized Driver Machine Interface (DMI) for all European trains.
- Reduction of trackside signalling equipment.
- Creation of a single radio communication system, which would support the new command-control system. The new radio system would also replace all legacy voice communication radios, e.g.: train-to-ground radio, tunnel radio, shunting radio, etc. [14]
- Introduction of Railway Emergency Call (REC) that offers fast and reliable communication in case of a dangerous situation.
- Cost reductions thanks to a single European market. Standardization of the command-control and communication system opens the local national markets to foreign competition. It also increases number of suppliers.

Features offered by ERTMS are not revolutionary when analysed separately. Actually, many of them were already available in the legacy control-command systems. Other ERTMS features were ideas taken from experimental systems, e.g. the French ASTREE [12, p. 27]. However, ERTMS, as a whole, is revolutionary, because it is the first system that brings all of these features together in a single, international standard.

ERTMS consists of two complementary elements: ETCS and GSM-R as shown in Figure 2.1. ETCS is a digital railway control-command system. It includes in-cab signalling, ATP system, standardized DMI, moving block and many more. The other ERTMS element is GSM-R. This radio communication technology has two main purposes. Firstly, it enables ETCS by offering data channels interconnecting trains and centralized control centres. Secondly, GSM-R is a unified solution for all railway voice communication.

ERTMS was initially developed in the EU for interconnecting the railway systems on the continent. The EU, via European Council Directives [15] and European Commission Decisions [16], obliged European railways to deploy ERTMS [12, p. 33]. However, due to many advantages of the system, other countries around the world also began to deploy it. Outside of Europe, ERTMS is used or planned to be used in: Algeria, Argentina, Australia, Brazil, China, Egypt, India, Indonesia, Kazakhstan, Libya, Malaysia, Mexico, Morocco, New Zealand, Russia, Saudi Arabia, South Korea, Taiwan, Turkey, and the United Arab Emirates [17]. Hence, ERTMS gradually

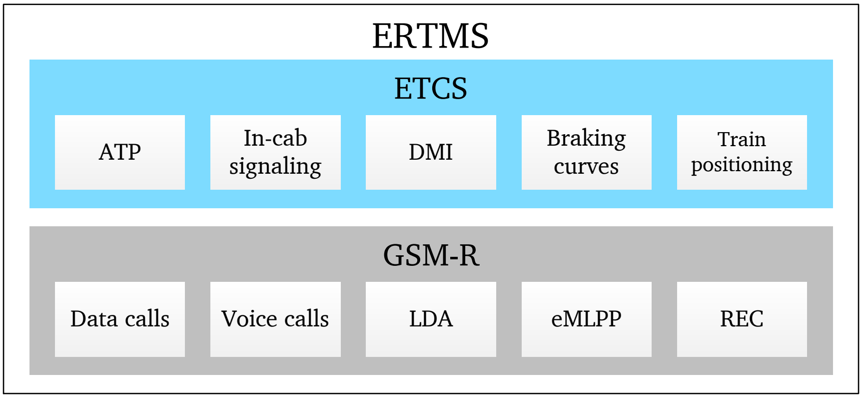


Figure 2.1: Elements of the ERTMS

turns from an European standard into a global standard with a broad support from countries around the world [18].

2.2 European Train Control System (ETCS)

2.2.1 Shortcomings of the classical signalling

The classical railway signalling, which is shown in Figure 2.2, is usually based on the colour light signals (earlier on semaphores). The usual (simplified) operation of the system is as follows [1, pp. 84–88]:

1. Firstly, when a train is scheduled to depart, the Traffic Management System sends a train route request to the interlocking.

2. The interlocking verifies whether the blocks (i.e. track sections) that will be included in the route are occupied or not. This is done via train detection system, such as axle counters.

3. Then, if the blocks are unoccupied, the interlocking sets the points (i.e. track switches) using point machines. Moreover, the interlocking verifies that respective signals display “stop” aspect, so other trains do not enter the route.

4. Once this is done, the appropriate signal aspect is displayed to the train driver.

This system ensures that two conflicting routes cannot be set up at the same time. Also, a “proceed” signal aspect guarantees that the route is locked [1, p. 61], i.e. all the points are in the correct positions, all the blocks are unoccupied and all conflicting routes receive a “stop” aspect. However, the safety of the system depends

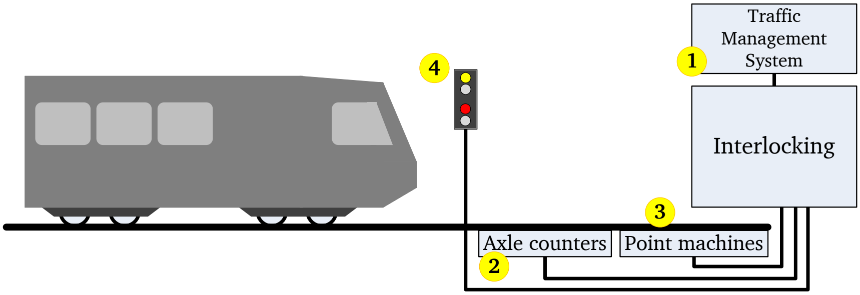


Figure 2.2: Classical railway signalling based on the colour light signals

significantly on the human factor, i.e. on the reaction of the driver to the displayed signal aspect [1, p. 208]. The driver may, for instance, fail to notice the signal, may misinterpret the signal aspect or may underestimate the stopping distance of the train. In case of such an error, the consequences may be fatal. Moreover, both the risk and consequence of the driver error increase with the train speed. Besides the safety concerns, the colour light signals have other drawbacks:

- They carry limited information [5, p. 18]. It is impossible to inform the driver about precise speed limits, track gradients and the exact distance to the “stop” signal location.
- They are located at fixed positions [5, p. 18]. Thus, if the speed limit is increased while the train is somewhere between two signals, the driver cannot be informed about it. The train will continue running at the old speed limit until the next signal becomes visible.
- They cannot take into account the characteristics of the particular train. For example, trains have different braking capabilities and maximum running speeds. Since the system does not know what train is currently running, the speed limits must assume the worst-case braking characteristics.

In order to address these shortcomings—therefore, to increase safety, efficiency, and capacity of the railway system—railways gradually move to computer-based signalling systems, such as ETCS.

2.2.2 Introduction to ETCS

ETCS, which is the state-of-the-art in railway command-control and signalling, is a communication-based system that manages and supervises train movement [4, 12]. It should be noted that ETCS does not replace the driver, so it does not provide Automatic Train Operation (ATO). However, the system supports the driver and reacts in case of a potentially dangerous error. The two main features provided by ETCS are: in-cab signalling and ATP:

- **In-cab signalling** is a concept that replaces the classical colour light signals with an interactive screen called the Driver Machine Interface (DMI) [4, p. 195]. The DMI, which is placed in the train cabin, displays all the commands and information necessary for the driver. Therefore, the risk that the driver may miss or misinterpret a signal is greatly reduced. Furthermore, information on the DMI is often more detailed and precise than information conveyed by the light signals [1, p. 180]. Hence, in-cab signalling contributes to both safety and efficiency of train operation. It is often a mandatory feature for trains running faster than 160 km/h [1, pp. 179–180].

- **Automatic Train Protection** is a system responsible for supervision of the train driver [1, pp. 208–212]. It is done by introducing an on-board computer that has knowledge about the speed limits, train characteristics and the safe stop location. Based on these, ATP calculates the braking curve that determines what should be the speed of the train at every point along the track [4, p. 100]. Then, by comparing this calculated speed with the actual speed and position of the train, the system verifies whether the driver obeys all the rules and whether the train can be stopped safely before it reaches the stop location. If the system detects that the train runs too fast or the driver does not react to the signals, automatic braking is applied [12, p. 20].

ETCS application levels

ETCS has three application levels: 1, 2 and 3 [1, p. 246]. All of the levels provide the in-cab signalling and ATP. However, they differ in terms of efficiency, investment cost and compatibility with the legacy signalling. Thanks to these multiple levels, the railways may adapt the system to their specific requirements and strategy. In Denmark, it has been decided to deploy ETCS Level 2 on the whole national railway network [19]. Therefore, the research work presented in this thesis considers ETCS Level 2. Consequently, in the following sections and chapters, ETCS Level 2 is referred to as ETCS.

ETCS replaces the way the driver receives signals (commands) from the system and it introduces elements supervising train movement. However, it does not replace the entire legacy signalling system, but builds on top of it. As shown in Figure 2.3, the interlocking, axle counters (or other track detection), point machines, and Traffic Management Systems are still necessary. However, these elements of the signalling system are out of scope of the ETCS standard and they differ from vendor to vendor.

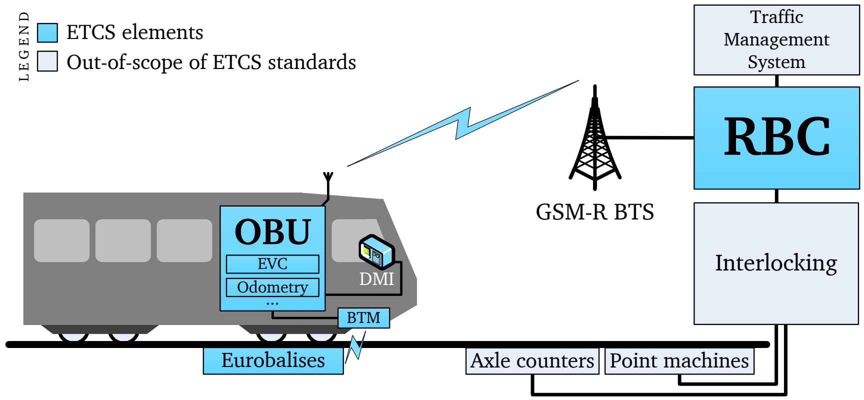


Figure 2.3: Schematic overview of ETCS Level 2 architecture

Architecture of the ETCS system

ETCS is divided into two general parts: on-board and trackside. The trackside part consists of the Radio Block Centre (RBC) and Eurobalises. The on-board part consists of the On-board Unit (OBU) and its supporting elements.

ETCS trackside elements

The *Radio Block Centre (RBC)* is the main element on the trackside part. It is a centralized computer that manages all trains running within its area [4, p. 49]. In Denmark, the entire railway network is planned to be divided into 36 areas, each supervised by a dedicated RBC [20, p. 18].

As illustrated in Figure 2.3, the RBC has interfaces with the Traffic Management System, the interlocking, and the train OBU. However, only the interface with OBU is standardized. The remaining two are proprietary solutions specific to a system vendor. Using these three interfaces, the RBC gets a detailed overview of the area it is responsible for. The Traffic Management System provides the current timetables—the up-to-date operational plan for each train. The OBUs provide information about the speed and position of the trains. The interlocking is responsible for setting and locking train routes—reserved and protected paths through the rail network.

The RBC manages train movement using Movement Authorities (MAs). An MA is a digital message containing a speed/distance envelope, i.e. a data vector defining the precise speed limits on the track section ahead of a train. Every MA includes the End of Movement Authority (EoMA), i.e. the stop location that the train is not authorized to pass until a new MA is issued [1, pp. 246–247].

Besides the RBC, the other trackside element is the *Eurobalise*, which is a transponder installed between the rails. While passing over an Eurobalise, a train receives a low-bitrate signal from it. In ETCS Level 2, this method is used to deliver static information, for example, about the precise position of the Eurobalise or about the RBC responsible for a particular area.

ETCS on-board elements

The *On-board Unit (OBU)* is a set of ETCS elements installed on the train. OBU consists of the European Vital Computer (EVC), DMI, Balise Transmission Module (BTM), an odometry system, and a GSM-R radio module [4, p. 32]. The EVC contains logic of the system, while the remaining elements provide interfaces and supporting functions, as follows:

- The DMI displays all commands and information necessary for a driver (e.g. speed limits, distance to EoMA). Thus, simplifying, DMI provides the functionality that was provided by the colour light signals in the classical signalling system. However, the DMI can also receive input from the driver, e.g. during ETCS setup procedure.
- The odometry system determines the current speed of the train and its position in relation to the last Eurobalise. Thus, it is an essential feature for ATP.
- The BTM reads the information sent by the Eurobalises that are placed along the track. Each Eurobalise sends its precisely defined position, which is used to correct the likely distance measurement error of the on-board odometry system [1, pp. 245–246].
- The GSM-R module provides a communication interface to the RBC.

The EVC interprets the MA messages incoming from the RBC (via GSM-R) and calculates safe braking curves. The braking curve defines the maximum speed that will still allow the train to stop before EoMA. As the train is running, the EVC controls if the driver follows the commands displayed on the DMI, i.e. the OBU controls if the train runs according to the issued MA. If the train speed approaches the braking curve, the EVC issues an audible warning. Then, if the driver does not react, and the braking curve is reached, an emergency brake is applied. In this way, ETCS provides ATP functionality that minimizes the risk of a human error.

2.3 GSM-Railways (GSM-R)

GSM-R is the first mobile communication standard designed specifically for railways. It is based on the Global System for Mobile Communication (GSM) standard, which is widely used in commercial mobile telephony networks. GSM-R provides two essential railway services [14]:

- Train-to-ground data communication for ETCS Level 2 and 3.
- Voice communication with specific features necessary for railways. The GSM-R network replaces train-to-ground radio, tunnel radio, shunting radio and maintenance radio, i.e. it is a single solution fulfilling all railway voice communication needs.

GSM-R is a network dedicated entirely to railways. This means that it is independent from other networks (e.g. commercial GSM networks) and it is not shared with entities other than railways (e.g. police or other public services). Also, GSM-R does not provide any services directly to the passengers, so their GSM terminals do not detect or connect to the GSM-R network.

2.3.1 Railways' choice of GSM-R

The work that eventually led to the development of GSM-R started in the late 1980s. At that time, concepts of a new communication-based signalling system started to emerge. These concepts later turned into ETCS, as described earlier in Section 2.2. However, already in the 1980s, it was foreseeable that future railways would need new mobile communication systems. Therefore, UIC initiated a discussion on reserving some of the GSM radio spectrum for the future railway use [12, p. 145].

Railways wanted to adopt a well-proven technology and use it for their purposes with a minimum of modifications [12, p. 145]. Two technologies were the strongest candidates: GSM and Terrestrial Trunked Radio (TETRA). GSM is a technology designed for commercial mobile telephony networks. TETRA is a network for public services, e.g. police, fire brigades, governmental institutions etc.

Both technologies had their advantages and disadvantages. GSM had large support from the telecommunication industry and a large base of suppliers. On the other hand, TETRA could provide bigger coverage and offered various features, which were useful for railways, e.g. group calls and direct mode operation (without infrastructure). However, in 1990, TETRA was still in the standardization process [21]. Therefore, GSM was chosen. The most important argument was that GSM had been an already proven technology with many products available on the market [12, 21].

2.3.2 Differences to the commercial standard

GSM was designed as a network for commercial mobile telephony. Therefore, the GSM standard had to be modified before it could be used in railway environment. There were several reasons for that [14]:

- Railway communication network must support users (trains), who travel with speeds up to 500 km/h [22, p. 27].

- Various services and applications delivered by the railway communication network have different importance and different impact on the railway safety. There is a need to differentiate between these services and provide them with various priorities in the network. Therefore, railway communication network must provide an efficient Quality of Service (QoS) mechanism.
- Railways require additional voice communication features, such as dynamic addressing and group calls.

In order to adapt the commercially used GSM standard for railways, the European Integrated Radio Enhanced NEtwork (EIRENE) project was initiated in 1992 by UIC [12, p. 145]. The goal of this project was to develop specifications for a GSM-based railway communication network. The project concluded in 1995 with publication of Functional Requirements Specification [23] and System Requirements Specification [22]. The EIRENE project was followed by Mobile Oriented RAdio NEtwork (MORANE) project, whose goal was to run three test GSM-R networks and validate the performance of the technology. This project finished in 2000, with a delivery of the final specifications of GSM-R [12, p. 146].

The principles of GSM-R are the same as GSM [12, p. 148]. Therefore, GSM-R provides all of the features of GSM. However, there are a few notable additions introduced in GSM-R by EIRENE and MORANE projects:

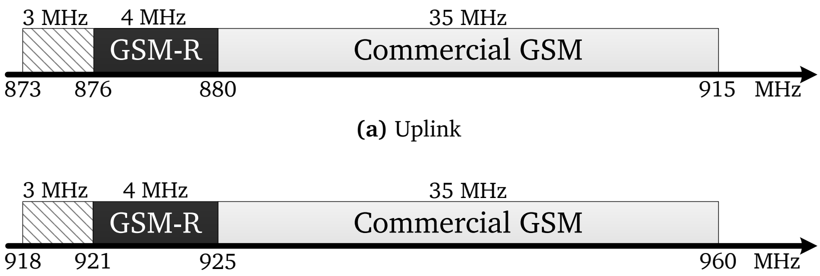
- *Enhanced Multi-Level Precedence and Pre-emption (eMLPP)* is a QoS mechanism that sets different priorities to calls and connections in GSM-R [22, 24]. For instance, eMLPP ensures that ETCS message exchange is not interrupted by a low-priority voice call. eMLPP is a necessary mechanism in a network where transmission resources are shared between safety-critical (e.g. ETCS) and other services.
- *Functional Addressing (FA)* allows users to call certain destination without knowing a specific phone number [14]. For example, it is possible to call a train dispatcher responsible for a given area, by simply pressing a single “Dispatcher” button on a GSM-R voice terminal. Another example is calling a train driver. Instead of knowing the particular phone number used by a train, it is possible to call the driver using the train running number.
- *Location Dependent Addressing (LDA)* dynamically selects the called party based on the caller location. This feature is used mainly when a train driver wants to connect with a dispatcher. LDA automatically chooses the dispatcher responsible for the given railway area. FA and LDA greatly simplify everyday railway operation and allow placing voice calls faster.
- *Voice Group Call Service (VGCS)* and *Voice Broadcast Service (VBS)* offer the possibility to make group and broadcast calls [24]. For instance, these features

may be used by a dispatcher to inform all train drivers about some disruption and the following travel delay.

- *Railway Emergency Call (REC)* is the most important GSM-R feature from the point of view of railway safety. REC is a high-priority broadcast call. It can be established from any GSM-R voice terminal using a dedicated REC button. REC pre-empts all ongoing voice calls and connects the caller with the dispatcher. All other terminals in the area automatically start to listen to the ongoing REC [22, p. 138]. Therefore, it is ensured that all railway personnel are immediately informed about the emergency situation.

Besides the additional features, another important difference to GSM is the dedicated radio frequency band in which GSM-R operates. Across Europe railways received an exclusive 4 MHz in 921 MHz radio band for GSM-R. As shown in Figure 2.4, 876–880 MHz is the uplink band, while 921–925 MHz is the downlink band [4, p. 148]. The common band used across the whole EU is one of the important elements allowing for cross-border interoperability.

In some countries GSM-R received an additional 3 MHz band: 873–876 MHz in uplink and 918–921 MHz in downlink. Thus, a total bandwidth of 7 MHz is available to GSM-R there.



(a) Uplink

(b) Downlink

Figure 2.4: GSM-R radio frequency bands in uplink and downlink. The hatched fields represent the additional band assigned to GSM-R in some of the European countries.

2.4 GSM-R principles and shortcomings

GSM-R network is divided into three main subsystems [12, p. 153]:

- *Mobile Station (MS)* is the user terminal attached wireless to the network. It may be a handheld voice terminal, a voice cab-radio or an ETCS OBU installed in a locomotive.

- *Base Station Subsystem (BSS)* is a Base Station Controller (BSC) and a number of Base Transceiver Stations (BTSs), managed by that BSC. BTS is a radio base station responsible for wireless communication with MSs.

- *Network and Switching Subsystem (NSS)* is commonly referred to as the “core network”. The most important nodes in NSS are: Mobile Switching Center (MSC), Home Location Register (HLR) and Visitor Location Register (VLR).

  MSC is the central element of NSS. It is responsible for management of the MSs (e.g. registration), call establishment, call routing and mobility management [25, p. 11].

Apart from the three subsystems above, GSM-R includes servers responsible for providing railway services (e.g. REC), as well as nodes responsible for operation and maintenance tasks (Operations and Maintenance Centre). The basic architecture of GSM-R is shown in Figure 2.5.

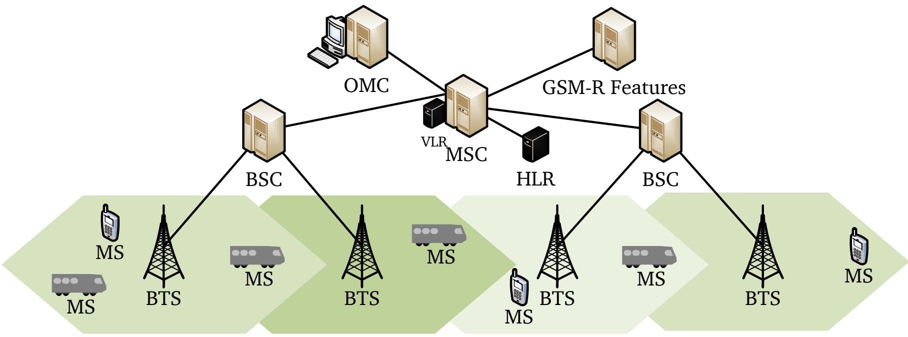


Figure 2.5: GSM-R network architecture

2.4.1 Main features of GSM-R

GSM was designed as a mobile network for providing telephony service (data communication was significantly less important) [25, p. 116]. Thus, technical solutions implemented in GSM were selected and optimized for that type of communication. The following paragraphs present the main features of GSM-R, while the next section discusses the consequences of the chosen solutions. This analysis is based on the previously published paper [Sniady2012b].

Cellular network

GSM-R is a cellular network, i.e. the connectivity with Mobile Stations (MSs) is delivered by a system of geographically distributed radio cells. The centre point of each cell is a BTS. From one side, the BTS provides radio coverage in its cell. On the other side, the BTS provides connectivity to the core network and the service offered there [25, p. 23].

Circuit-switched based transmission

GSM-R is a circuit-switched network. Therefore, every connection in the network (call or a data connection) requires a dedicated end-to-end virtual circuit [13]. This means that network resources are reserved exclusively for a particular connection, both on the radio and the backbone links.

Frequency-Division Duplex (FDD)

GSM-R is an FDD technology, so the uplink (from an MS to a BTS) and downlink transmission are carried on separate frequencies, as shown previously in Figure 2.4. Hence, the 4 MHz GSM-R band consists actually of a 4 MHz band in uplink and a 4 MHz band in downlink [4, p. 148].

The uplink and downlink resources are assigned symmetrically, in a sense that an active connection always receives equal uplink and downlink network resources. In the following paragraphs, only one direction is discussed, but the description applies equally to both of them.

Frequency channels

The 4 MHz GSM-R radio band is divided into 19 frequency channels, each being 200 kHz wide [13]. These channels are used to separate transmissions in neighbouring cells. Hence, the frequency channels must be distributed among cells in a way ensuring that neighbouring cells do not use the same frequencies (the same channels). Each radio cell uses one or more channels depending on the expected capacity demand.

A frequency channel that is used in one cell can be reused in another cell, but only if the distance between them guarantees that the cells do not interfere with each other. Seven frequency channels are usually required to provide coverage over a wide area. On an open railway line, where only a linear coverage must be provided, four channels may be sufficient [26, pp. 103–106]. Since the GSM-R band offers 19 channels, cells are assigned sets of channels instead of a single channel. For instance, if seven sets are defined, then each of them includes two or three channels. Therefore, two or three frequency channels are available in each cell. An exemplary channel distribution is presented in Figure 2.6.

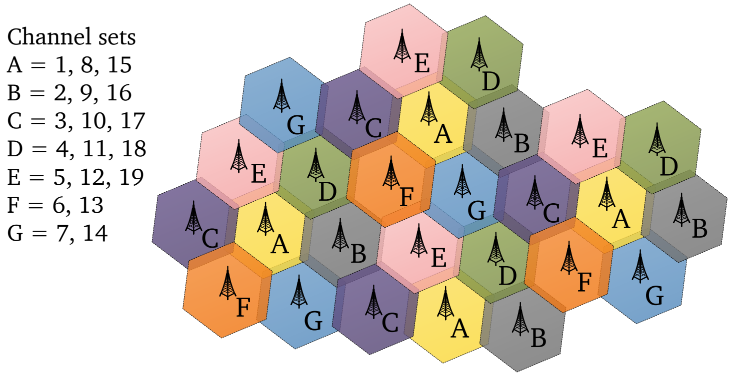


Figure 2.6: Example of frequency channel distribution with seven channel sets. Each set is marked with a letter and a distinctive color.

Time Division Multiple Access (TDMA)

In order to provide multiple calls (circuits) per cell, each frequency channel is shared between MSs using TDMA. GSM-R radio transmission is divided into frames. A frame lasts 4.615 ms and consists of eight time-slots (each 0.577 ms), as shown in Figure 2.7. In each cell, at least one time-slot is reserved for network signalling [25]. The remaining seven slots carry user calls/connections, i.e. the virtual circuits.

A GSM-R call occupies one TDMA time-slot in every consecutive frame. This time-slot is reserved exclusively for the call, until the call is finished. Since seven time-slots are available, seven simultaneous calls can be carried over a single frequency channel (assuming that one time-slot is used for network signalling). Inactive MSs do not occupy any time-slots. As explained earlier, a cell usually offers up to three frequency channels and each of them carries seven connections. Therefore, capacity of a typical cell is 23 connections, i.e. traffic channels. These capacity considerations have been published in [Sniady2014c].

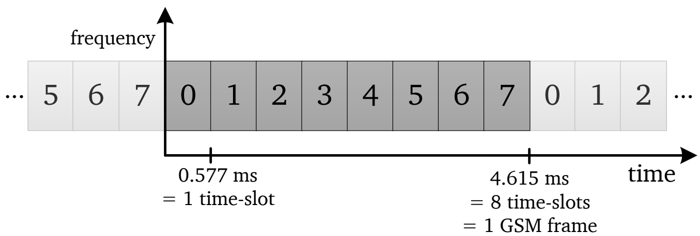


Figure 2.7: GSM-R TDMA frame divided into eight time-slots

Radio modulation

Gaussian Minimum Shift Keying (GMSK) modulation, which is used in GSM-R, was chosen due to its simplicity in hardware implementation and low interference emission. However, regardless of the radio conditions, it transmits only one bit per symbol [25, pp. 43, 70].

2.4.2 Consequences of the GSM-R design choices

GSM standard was designed taking into account two important assumptions that affect the performance and capabilities of GSM-R networks today:

- GSM network will be used predominantly for voice service [25, p. 116].
- MSs will offer little computing power and limited battery life.

Nowadays, these assumptions do not hold. This is because, since the early 1990s, when GSM was designed, both the communication demands and the capabilities of electronic devices have evolved significantly.

First and foremost, data communication is now equally or even more important than voice communication. In railways, this was already true when GSM-R was being designed as a technology supporting ETCS command-control system. ETCS is based on data communication. Moreover, new communication-based applications and services are foreseen for railways (see Section 1.2). Hence, a modern railway communication network must provide good support for data transmission.

Secondly, thanks to the advances in electronics, the computational power of modern mobile terminals allows implementing much more advanced modulation and multiplexing solutions [25, p. 217]. Also, the battery life is usually less of an issue in the railway environment, because many terminals (MSs) have continuous power supply, e.g. from a locomotive.

Despite these significant changes, the GSM-R standard remained unmodified. Some of the design choices that made GSM-R a good technology for voice communication in the 1990s became its shortcomings today:

- GSM-R does not provide packet-switched based transmission. Therefore, data communication must be delivered by Circuit-Switched Data (CSD), which cannot assign the network resources based on the actual demand. This means that data is transmitted over virtual circuits, just like voice frames [27]. However, in opposite to voice, data communication is most often bursty. Data source sends varying amount of data at irregular intervals. Such a bursty transmission does not fit well into a fixed circuit provided by GSM-R. As a result, circuits are often underutilized and network resources are wasted [28].

- TDMA assigns one time-slot a frame to each connection. This fits well with voice encoders that encode speech into periodical frames. Bursty data connections, such as ETCS, could benefit from using more time-slots per frame. However, a single connection cannot get more than one time-slot, even if spare time-slots are available. Therefore, the radio resources may stay unassigned even if there is data traffic waiting for transmission [25, p. 64].

- GSM-R resources are assigned symmetrically in uplink and downlink. However, data-based services often generate different amount of traffic in the two directions [25, p. 67]. Hence, symmetry of GSM-R connections means that either uplink or downlink is overbooked and the network resources are wasted further.

- GMSK modulation scheme, which is used in GSM-R, is unable to take the full advantage of good radio conditions. GMSK is sufficient for voice communication, but more advanced modulations schemes would allow GSM-R to transmit at much higher bitrates [25, p. 70]. Thanks to the advances in electronics, nowadays even handheld devices are capable of using more advanced multiplexing and modulation techniques [25, p. 217], such as Orthogonal Frequency-Division Multiple Access (OFDMA) and Quadrature Amplitude Modulation (QAM).

- The maximum connection bitrate offered by GSM-R is only 9.6 kbit/s [22, p. 38]. This is a consequence of many design choices such as the modulation and multiplexing schemes. Such a low bitrate is insufficient for many modern applications, especially those based on transmission of multimedia.

- Transmission latency in GSM-R network is estimated to be in the range between 200 ms [29] and 400 ms [4, p. 162]. If the low bitrate is added to that, the GSM-R delay performance turns out to be very poor. Thus, GSM-R may not fulfil requirements of delay-sensitive applications.

- GSM-R Call setup time is in the range of about 5 s [4, p. 162]. GSM-R requirements state that the setup procedure cannot exceed 8.5 s (95% of cases) and 10 s (100% of cases) [30]. This may be sufficient for a voice call, but such a long connection setup time is unacceptable for many real-time applications.

GSM-R is inefficient as a data network

The inflexible mechanisms implemented in GSM-R cannot adapt to the varying demands of data-based applications. Data connections are provided over CSD calls, which offer a fixed constant bandwidth regardless of the actual traffic load. Therefore, network resources are overbooked and underutilized. For example, an ETCS connection may use as little as 0.5% of the resources it receives [4, p. 155].

GSM-R capacity is insufficient

The inefficiency and inflexibility of GSM-R have another consequence. In order to deliver ETCS data, a GSM-R network must establish a dedicated CSD call to each train. Thus, each ETCS connection (CSD call) exclusively occupies one of the few available GSM-R time-slots.

In case of voice communication, allocation of network resources exclusively to a single call is not problematic, because voice calls are usually short. Besides, all users do not call at the same time. However, in case of ETCS, all trains must be continuously and simultaneously connected to the RBC. For this purpose, each train must have a continuous CSD call established. The call lasts as long as the train is operating [4, p. 154]. Therefore, a GSM-R cell can only accommodate as many trains as many traffic channels (time-slots) it has available. As it was explained in Section 2.4.1, a typical cell offers 23 traffic channels. Since each train occupies one channel, such a typical cell can accommodate at most 23 ETCS-equipped trains. However, in practice, some of the traffic channels must be kept for voice communication, as well as for handover procedures [19, p. 4]. Therefore, a typical GSM-R cell can accommodate less than 20 trains.

It has been widely recognized that GSM-R capacity is insufficient, especially in areas with high density of train traffic [4, 13, 31]. This means that the capacity of a GSM-R cell becomes a bottleneck limiting the number of trains operating in a given area. This is very undesirable, because the only limitation should be due to the available rail infrastructure and not due to telecommunication infrastructure.

GSM-R is obsolete

The railway communication demand is continuously increasing, especially in terms of data transmission capacity. It is expected that this process will continue [9] [11, p. 40]. As this is happening, it becomes apparent that GSM-R is an obsolete technology. Due to its inflexibility, inefficiency, and limited capacity, GSM-R will not be able to answer the communication needs of the future railways.

In commercial mobile networks, the shortcomings of GSM were noticed and acknowledged long time ago. Therefore, the telecommunication industry developed a number of newer standards that addressed these shortcomings. GSM was succeeded by General Packet Radio Service (GPRS), Universal Mobile Telecommunication System (UMTS), High Speed Packet Access (HSPA) and the most recent family of LTE standards [25]. This evolution is illustrated in Figure 2.8. Railway communication technologies, on the contrary, did not follow the commercial standards and kept GSM-R mostly unchanged.

In commercial mobile telephony networks, the number of subscribers using GSM-only terminals is predicted to decrease at an average rate of -15% per year [32, p. 30]. Thus, the importance of GSM technology in commercial mobile networks

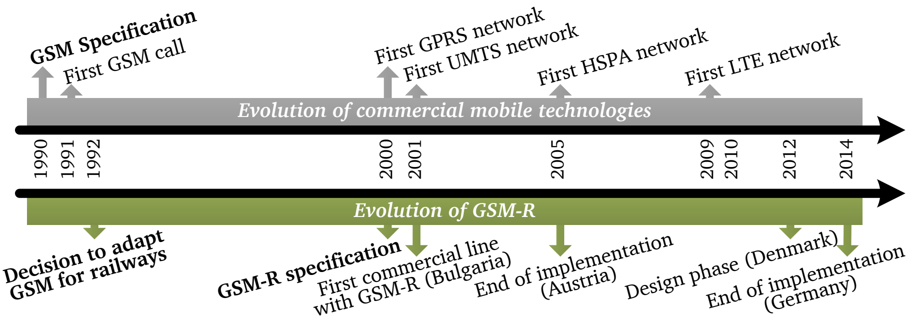


Figure 2.8: Evolution of commercial and railway mobile communication

is expected to decrease as well. This process will lead to gradual decline in the industry support for GSM and then GSM-R [11, p. 40]. Afterwards, the installation and maintenance costs may grow as the supplier base decreases. Due to that, in a recent report prepared for the European Railway Agency (ERA) [33, p. 7], it is predicted that the railway industry will not be able to support GSM-R after 2030.

2.5 Future alternatives

GSM-R capacity and data transmission capabilities are insufficient considering today's railway communication needs. In the future, GSM-R shortcomings will become even more problematic as demand for communication-based services will grow. Therefore, railways seek various technical and operational solutions to these shortcomings.

In Denmark, the problems caused by the likely communication capacity constraints of GSM-R are expected to be lessened by special operational rules, such as [19, p. 4]:

- Train dispatchers should supervise the number of trains in each area and ensure that free GSM-R traffic channels are available for the incoming trains.
- Drivers of trains at standstill may be required to shut down their ETCS systems in order to release the network resources.

These solutions only reduce the consequences of the insufficient capacity, but they do not solve the problem. Besides, due to them, the work procedures for drivers and dispatchers are unnecessarily complicated. For instance, after shutting down the ETCS system, in order to drive again, the driver will be required to go through the whole ETCS start up procedure.

Another solution is to allocate additional frequency bands to GSM-R. In some countries, e.g. in Germany, an additional 3 MHz frequency band is given to railways. GSM-R uses also the 873–876 MHz and 918–921 MHz bands there, as shown in Figure 2.4. This provides space for additional frequency channels. However, in other European countries, these bands are already reserved for other purposes [31].

These solutions are temporary, because they do not address the core of the problem, i.e. the shortcomings of the GSM-R standard. Moreover, they only try to address the most urgent capacity issue. They cannot improve data transmission capabilities, increase bitrates or lower the transfer delay. Therefore, railways need an alternative technology that would solve all of the GSM-R shortcomings and that could replace GSM-R in the future.

European Railway Agency (ERA) started initial studies on the evolution of railway mobile communication. The agency set a goal of defining a new communication system, together with a migration strategy by 2018. The system is supposed to be ready for deployment by 2022 [34].

Various technologies have been considered by researchers, telecommunication system suppliers, and railway companies as candidates for the future railway communication network. All of the proposed technologies are well-proven telecommunication standards. This is due to the railway choice to reuse available technologies and architectures with as little modifications as possible [12, p. 145]. Such a strategy should increase competition between railway telecommunication suppliers and eliminate vendor lock-in [8].

The following sections present the most notable technologies that have been proposed to become the future railway communication technology.

2.5.1 Terrestrial Trunked Radio (TETRA)

In EU, railways are required to deploy ETCS system based on GSM-R communication network. However, outside Europe, where EC Directives [16] do not apply, railway companies consider alternative technologies [18].

TETRA, which was rejected by the European railways in 1990s', eventually became a railway communication technology in Kazakhstan [35] and in Taiwan [11, p. 20]. The main reasons for choosing it over GSM-R were lower cost, wider bandwidth (higher capacity) and longer range of TETRA cells.

However, apart from the lower cost, TETRA would not bring any significant improvements over GSM-R. Since TETRA offers data transmission rates only up to 28.8 kbit/s [36, p. 135], it would not solve the main issue of GSM-R, i.e. the limited support for data communication.

2.5.2 General Packet Radio Service (GPRS)

General Packet Radio Service (GPRS) is an extension of GSM standard. It introduces end-to-end packet-switched transmission. GPRS adds more flexibility to the radio interface thanks to time-slot aggregation. Moreover, the offered data rates are higher due to new coding schemes with lower redundancy. This allows network to take better advantage of good radio conditions (e.g. in an area close to a base station) [25, pp. 63–70].

These improvements makes GPRS much more efficient in data transmission than the original GSM (and GSM-R). In the railway context, it means that ETCS connections would be transmitted in packet-switched mode. They would use radio and backbone resources only when an ETCS message is actually transmitted. The network resources in GPRS would be efficiently shared between the ETCS flows. Therefore, the capacity of the network, in terms of concurrent ETCS connections would increase [13]. Another advantage of GPRS is that it supports simultaneously GSM and GPRS connections [25, p. 68]. Thus, an old GSM/GSM-R MS can still use the network without any upgrades. This is a considerable advantage, since it would ease the migration process. Finally, upgrading a GSM-R network to GPRS is much simpler and cheaper than deploying some other technology.

Although GPRS was already standardized in the mid-1990s [28] and railways have already been considering this technology for many years [4, 12, 19], GPRS is still not approved as a communication technology for ETCS [11, p. 5]. However, railways seem to agree that GPRS is the next step in the evolution of the railway mobile networks. This is why Banedanmark, the Danish railway infrastructure operator, decided that ETCS system in Denmark should already be based on a GPRS-enabled GSM-R network [19].

Even though GPRS may solve the most urgent capacity issues of GSM-R, its performance in terms of delay and offered data capacity is still relatively poor. The transfer delay in GPRS is worse than in GSM-R CSD calls [13], especially under high network load [4, p. 163]. Besides, handover delay in GPRS is in the range of seconds [13], which may be problematic for fast running trains. These delay issues are of big concern for railways [12]. Moreover, GPRS data transmission capabilities are still much worse than those of newer standards. Also, telecommunication industry support for GPRS is going to decrease together with GSM support. Thus, due to these shortcomings, GPRS is likely only a temporary solution, until a next generation communication technology for railways is defined.

Nevertheless, deployment of GPRS as a technology for delivering ETCS has a very significant "side-effect". Since GPRS is a packet-switched technology, railways must update ETCS specification in order to allow ETCS communication over packet-switched IP-based networks [11, p. 6]. Hence, ETCS signalling will become independent from the underlying technology, as explained further in Section 3.2.

This is a very important step, because it will allow ETCS connectivity to be provided using any IP-based technology that fulfils certain performance requirements.

2.5.3 Universal Mobile Telecommunications System (UMTS)

Universal Mobile Telecommunication System (UMTS) is a successor of GSM and GPRS. The first release of UMTS standard (Release 99) introduced a more advanced radio interface. In the following releases of the standard (Release 4) the backbone network evolved as well [25, pp. 116–119].

Idea of using UMTS for railway command-control systems gained little interest in Europe. However, it is used in Australia. It is worth noting that Australian railways take advantage of various technologies in order to provide connectivity with their trains. Apart from UMTS, also GSM-R, Ultra High Frequency (UHF) and satellite communication are used [11, p. 70].

UMTS offers many improvements over GSM-R. However, there are already newer standards available, which provide even better performance and, at the same time, have much lower obsolescence risk than UMTS. Since deployment of a new communication technology for railways will be a complicated and costly process, the chosen technology should be the state-of-the-art among telecommunication standards. Hence, UMTS is unlikely to be chosen as the successor of GSM-R.

2.5.4 802.16 Worldwide Interoperability for Microwave Access (WiMAX)

Worldwide Interoperability for Microwave Access (WiMAX) has also been proposed as a candidate for the future GSM-R replacement [8]. The technology has notable advantages that could make it a good choice for railway communication:

- WiMAX is a packet-switched technology offering large bandwidth and efficient data communication. Realistic data rates reach 16 Mbit/s, while theoretical ones reach 78 Mbit/s [25, p. 284].

- Radio interface in WiMAX is based on Orthogonal Frequency-Division Multiplexing (OFDM), which is significantly more efficient and flexible than the TDMA mechanism used in GSM-R.

- WiMAX supports a range of modulation schemes (up to 64QAM), which can be dynamically chosen based on the radio conditions.

- WiMAX offers group-calls and push-to-talk voice communication [8].

- Multi-hop mesh networking is available, which could be used as a cost-effective method for connecting remote base stations.

- WiMAX also offers QoS mechanism for prioritizing different flows in the network, which is important if the transmission resources are shared between critical and non-critical applications.

Telecommunication industry initially had a large interest in WiMAX standard. It was one of the main candidates to become the fourth generation mobile communication technology. However, WiMAX did not achieve a large commercial success. The currently deployed WiMAX networks are usually small and they serve private institutions, local communities or small towns. Hence, WiMAX gained only about 25 million subscribers globally [37, p. 4].

From the railway perspective, the low popularity of the technology is a major drawback, since WiMAX cannot guarantee a long-term wide industry support or a large vendor base. Therefore, choosing WiMAX as the future railway communication network, railways may end up with a niche technology, which will turn out to be very costly in maintenance.

Nevertheless, the benefits offered by WiMAX may be still available to railways. This is because, as explained in the next section, LTE includes many technical solutions similar to WiMAX (e.g. modulation), but has significantly larger industry support. As a consequence, LTE is a more likely choice for railways.

2.5.5 Long Term Evolution (LTE)

Long Term Evolution (LTE) is the latest family of mobile communication standards developed by Third Generation Partnership Project (3GPP). LTE is an indirect successor of GSM. As such, it is a natural candidate for GSM-R replacement.

The first release of the LTE standard was published almost 20 years after the GSM standard. LTE is a result of many developments and advances in telecommunications and electronics that occurred during these years. Thus, from the telecommunications point of view, there are huge differences between these two mobile technologies.

All the evolution from GSM to LTE was motivated by the needs of commercial mobile networks. The new solutions are optimized for this type of networks. However, most of them may be equally important for railway mobile networks. The most notable advantages of LTE that may be beneficial for railways are as follows:

- LTE is the first fully packet-switched IP-based mobile communication standard from 3GPP [25, p. 206]. In opposite to the previous standards, in LTE, both data and voice communication is packet based. Compared with GSM-R, LTE network assigns the network resources to users and applications depending on the actual transmission demand.

- LTE introduces a simplified backbone network called Evolved Packet Core (EPC) with fewer elements than in the legacy standards. The circuit-switched part of the backbone, which was used in earlier network generations for voice transmission, has been abandoned in favour of a fully packet-switched solution. Thanks to this choice, EPC may use any IP-based transport network [25, p. 211], such as Carrier Ethernet.

- LTE introduces a new radio interface based on Orthogonal Frequency-Division Multiple Access (OFDMA) in downlink and Single-carrier FDMA (SC-FDMA) in uplink [25, pp. 218–222].

- Modulation and coding schemes are dynamically chosen in LTE based on the radio conditions and the traffic demand [38, p. 217]. This link adaptation mechanism allows the network to balance between throughput and reliability of the radio transmission.

- The new radio interface offers much higher spectral efficiency than any other legacy mobile communication standard [39, pp. 242–244]. This is due to the advanced modulation (OFDMA), multiplexing (up to 64QAM) as well as usage of Multiple Input Multiple Output (MIMO).

- LTE can operate in different bandwidths: 1.4 MHz, 3 MHz, 5 MHz, 10 MHz, 15 MHz or 20 MHz (and more with carrier aggregation in LTE-Advanced). This range of bandwidths allows network operators to flexibly manage their available radio spectrum. For instance, an operator may split the radio band used by the GSM network (spectrum refarming). One part would still be used by GSM, while in the other part a new LTE network would be deployed. By selecting a wider or a narrower LTE bandwidth, the network operator may decide how big a part of the radio resources to assign to LTE and how much to keep for GSM [39, p. 246].

  This may be a very important feature for railways, because it means that the railway LTE network could be deployed along the GSM-R network in the same band. As the number of terminals equipped with LTE radio would increase over time, the bandwidth of the LTE network could be increased accordingly. Therefore, the migration to LTE would be gradual and spread over years.

- LTE offers QoS mechanisms providing traffic differentiation, protection and prioritization over both radio and backbone networks.

- LTE provides standardized mechanisms for inter-working with all legacy 3GPP technologies (e.g. GSM). Mechanisms, such as cell re-selection, handover and connection release, allow mobile terminals to quickly and seamlessly transfer from one radio network to another [25, pp. 254–258]. Thanks to them, migration from GSM to LTE may be gradual and interoperability between the new and old systems can be provided.

- LTE is the latest family of mobile communication standards. Hence, it has much lower obsolescence risk than any of the previous standards (e.g. GPRS, UMTS or HSPA).
- LTE gained wide support from the telecommunication industry. The LTE deployments are continuously growing. In North America, Europe, South Korea and Japan, it is predicted that LTE terminals will constitute over 50% of the total mobile subscriptions by 2019 [32, p. 9]. Therefore, long term industry support for LTE can be expected.

All in all, LTE is an efficient technology, which offers high transmission capacity and low latency. Usage of the network resources—especially the limited radio resources—is optimized in comparison to GSM-R. Moreover, the transmission performance is significantly improved in terms of throughput and delay. Also, LTE offers standardized solutions for inter-working with GSM.

Thanks to the above characteristics, LTE is gaining a considerable interest from the research community, railway industry and suppliers, as one of the most likely candidates for GSM-R replacement in the future [10, 11, 40–44].

2.6 Research motivation and goals

Due to the shortcomings of GSM-R, railway industry began the process of choosing a new communication technology. As it was explained in the previous section, from the range of mobile communication standards, LTE is a likely candidate to become the successor of GSM-R.

Related research on LTE in railways

The idea of using LTE instead of GSM-R emerged in the railway and the telecommunication industries already before 2010 [43] [44, p. 30]. However, the research and the available publications on this topic have been limited.

In the related literature, the main area of interest was LTE performance in high-speed scenarios, which are challenging due to the Doppler shift [45] and frequent inter-cell handovers [46]. In one of the first works on LTE in railway environment, Luo et al. [47] investigated LTE handover procedure and proposed a method for a dynamic reconfiguration of handover parameters depending on the user (e.g. train) speed. Feng et al. [48] considered the issue of channel estimation and Inter-Carrier Interference (ICI) under high-speed conditions. LTE connectivity for train on-board users is still a popular research topic, as proven by many recent publications proposing to address this challenge with novel solutions, such as multi-cell connectivity [46], improvements in the physical random access channel [49] and fixed-point handovers [50].

Nevertheless, in all of these published works, LTE is considered as a commercial network for private customers—connected directly or via on-board relay nodes. Only a few researchers analysed LTE as a network supporting critical railway applications, which is the main role of the GSM-R network. Liem and Mendiratta [41] compared LTE and GSM-R in terms of availability of voice communication. They concluded that LTE can offer an improvement compared to the current railway networks. Then, Calle-Sanchez et al. [40] explicitly considered LTE as an ERTMS-supporting technology and VoLTE as a likely solution for providing railway voice communication. However, the authors presented only a theoretical analysis, which did not include any performance evaluation.

Research on LTE for CBTC-based railways is also worth noting, because it considers LTE as a network supporting a train command-control system—similarly as LTE would support ETCS in ERTMS-based railways. Gresset et al. [51] presented a QoS-aware scheduler for maximization of radio throughput in an LTE network for CBTC. Khayat et al. [52] discussed LTE QoS configurations and their impact on transmission of CBTC messages. However, both of these publications concerned urban railway systems, which have very different communication requirements—in terms of bandwidth, delay, command-control traffic, cell size, user speed, etc.—than ERTMS-based railways.

Goal of this research work

Considering its advantages, LTE should be able to address all major GSM-R short-comings, i.e. the insufficient capacity, the inefficiency in network resource usage and the limited support for data communication. Although usage of LTE for railway was considered, none of the research publications explicitly confronted LTE with railway communication requirements, especially those defined by the ETCS system. In the two publications proposing LTE as a possible successor of GSM-R [40, 41], only theoretical considerations on railway voice communication in LTE were presented.

Despite its well-known advantages, LTE is a technology designed for commercial mobile networks that provide data and voice transmission for a variety of information, entertainment and other non-critical applications. Hence, LTE has not been intended for a very specific and niche networks such as the railway network. This disparity between the original purpose of the LTE standard and its possible usage in railways is the principal motivation for the research presented in this thesis. Therefore, the main question that this thesis tries to answer is: can LTE become the future railway communication technology? In order to answer this general question, the following more particular goals have been identified:

- Validate LTE in scenarios that explicitly model railway environment and its specifics, such as user concentration at the stations and along railway lines, user mobility and speed, power constraints, radio propagation conditions (urban, rural), etc.

- Analyse LTE as a network providing the necessary communication for the ETCS command-control system. Investigate transmission performance experienced by ETCS traffic under various conditions: user speed, radio base station density, and traffic load.

- Analyse LTE as a network for railway voice communication and verify whether VoLTE standard is able to provide essential railway features, such as the Railway Emergency Call (REC).

- Compare the transmission performance offered by LTE with railway communication requirements, e.g. in terms of transfer delay, data integrity and call setup times.

- Verify whether the QoS mechanism offered by LTE is able to provide sufficient prioritization and differentiation between safety-critical real-time railway applications (e.g. ETCS) and non-critical applications, e.g. passenger voice announcements and video surveillance.

- Propose and validate alternative radio network architecture for increased resilience, availability and capacity of the railway communication network.

The next three chapters present outcomes of the research work that addressed the challenges listed above. Chapter 3 discusses LTE as the communication technology for ETCS signalling and other data-based railway applications. Then, Chapter 4 investigates usage of VoLTE for providing railway voice communication. Finally, Chapter 5 considers heterogeneous LTE deployments and methods for providing data integrity.

CHAPTER 3

ETCS signalling in LTE

The European Train Control System (ETCS), which is a digital computer-based train signalling system, is an essential element of modern railways. ETCS offers many features that significantly increase safety, efficiency, and cost-effectiveness of railway operation. Among other things, the system provides: in-cab signalling, Automatic Train Protection (ATP), and moving block operation. Thanks to these features, trains can run at high speed, the track capacity increases, and the risk of a human error is minimized. ETCS is one of the main reasons why railways need a supporting mobile communication network. Therefore, the principal criteria for evaluating LTE as a possible railway communication technology are as follows:

- What is the transmission performance that LTE can offer to the ETCS system? How is this transmission affected by deployment strategy (in terms of base station density), train speed, and traffic load?
- Does LTE fulfil ETCS requirements—in terms of transfer delay and data loss probability—under varying conditions?
- Can LTE provide both critical and non-critical applications over shared radio resources?

Chapter organization

This chapter is organized as follows. Section 3.1 explains the importance of mobile communication for the ETCS system. Section 3.2 describes migration of ETCS to networks based on Internet Protocol (IP). Section 3.3 proposes LTE as a communication technology for ETCS support. Then, the remainder of the chapter investigates ETCS transmission performance in an LTE network under various conditions based on realistic railway scenarios.

3.1 Role of communication in the ETCS system

ETCS is a distributed system that has its main logic split between two elements: the OBU and the RBC. This is illustrated in Figure 3.1. As it was explained in Section 2.2 on page 10, there are more elements in the system. However, since their operation does not depend directly on the underlying mobile communication technology, they are out of scope of this work.

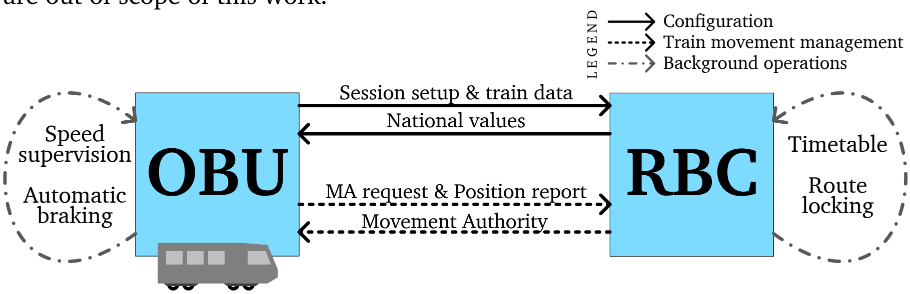


Figure 3.1: Information exchange between ETCS elements. The arrows are a simplified illustration of the information flows in the system, but they do not show the exact ETCS message exchange.

In order to start the ETCS operation, i.e. train movement supervision, an ETCS session must be established between the OBU and the RBC. During the establishment process, the two ETCS elements exchange configuration parameters which are necessary for the later operation. For example, the OBU sends train data (e.g. train running number, train length, driver ID, etc.). In the other direction, the RBC sends the national values (e.g. the default speed limits, reverse movement protection, etc.) [53, p. 34]. The session establishment is the first situation when the RBC and the OBU must communicate over a mobile network.

The RBC is responsible for management and supervision of train movement. In order to do that, the RBC must have an up-to-date overview of its area regarding both static and dynamic information [4, p. 49]. The fundamental part of this overview are the train positions. Therefore, each OBU repeatedly sends a position report message to the RBC. Most often, the report is joined together with a Movement Authority (MA) request [4, p. 107].

Based on the position report and other information (timetables, interlocked routes, other train positions), the RBC issues an MA for the train. The MA defines speed limits and distance that the train is allowed to drive. As the train is running, the position reports and the MAs are repeatedly updated and exchanged between the OBU and the RBC.

Since the ETCS system is based on this real-time cooperation between the RBC and the OBU, a continuous communication between these two elements is a necessity. Without the RBC-OBU connectivity, the core functionality of ETCS could not be provided, i.e. the MA distribution, the in-cab signalling and the ATP.

The OBU-RBC communication must fulfil certain quality requirements, especially during particular ETCS procedures, such as the *MA extension*. If a train approaches the End of Movement Authority (EoMA) and the updated MA cannot be delivered, then the train is forced to slow down and eventually to stop. The OBU cannot allow the driver to pass the EoMA unless the updated MA arrives. Therefore, a disruption or a long delay in the OBU-RBC communication may cause a train travel delay, which may then propagate through the railway system and cause further knock-on delays for other trains [54]. This may also reduce the available track capacity [53, pp. 23–31].

This example of the MA distribution illustrates how the QoS of the underlying communication network impacts the QoS of the ETCS system. This, in turn, impacts the QoS of the railway operation (e.g. in terms of travel delays) [53, p. 13]. Hence, in ETCS railways, an efficient and robust data communication is one of the foundations of a reliable, efficient, and punctual train service.

Nevertheless, it should be pointed out that despite the importance of the underlying communication, the safety of the railway operation relies entirely upon the ETCS system [11, p. 51]. Even in the event of a complete failure of the OBU-RBC communication, the ETCS procedures must ensure that trains cannot make any potentially dangerous movements. It means, in practice, that trains are preventively stopped until the communication is restored and the RBC reacquires the up-to-date overview of the whole area.

3.2 ETCS migration to IP-based networks

Currently, the OBU-RBC communication is provided by the GSM-R network, as illustrated in Figure 3.2. The OBU is connected over a radio link, while the RBC is connected over a fixed link. Usually, the fixed part of the GSM-R network is based on the Integrated Services Digital Network (ISDN). More details on GSM-R have been presented in Section 2.3 on page 14.

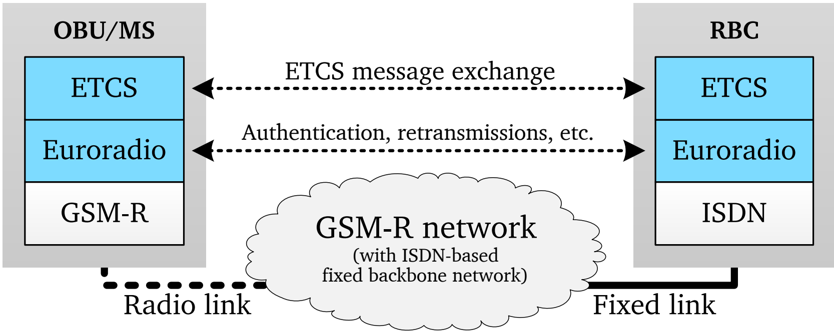


Figure 3.2: OBU-RBC communication based on the GSM-R network

As visible in Figure 3.2, between the ETCS application layer and the underlying GSM-R there is an additional protocol layer that is called *Euroradio*. Its role is to provide authentication, data segmentation and reassembly, error-checking, data loss detection, and data retransmission. Hence, Euroradio includes all the necessary mechanisms for ensuring a reliable end-to-end ETCS data transmission even in case of errors in the underlying radio network [4, pp. 56–162].

Both ETCS and Euroradio have been designed under the assumption that the OBU-RBC connectivity is provided specifically by the GSM-R network. Due to that, Euroradio interfaces are defined only with a GSM-R or an ISDN network [55, pp. 6–16]. Furthermore, packet-switched transmission is explicitly excluded from the current specifications [55, p. 14].

However, it seems to be broadly agreed in the railway industry, that the future railway communication technology will be packet-switched based—more specifically, will be based on the IP [11,33,34]. Migration to IP will require ETCS and Euroradio specifications to be redefined and adapted to the IP protocol stack [4, p. 163]. Once this is done, ETCS functionality will become decoupled from the underlying network. Therefore, ETCS will become a future-proof standard that will be independent from the particular technology providing the OBU-RBC connectivity.

As mentioned in Section 2.5.2 on page 26, some railway companies are already considering an introduction of GPRS in order to increase the data transmission capacity. Since GPRS is an IP-based technology, the work on migrating ETCS to the IP protocol stack has already started [11, p. 49]. The move to IP will not only allow railways to use GPRS, but also—in the future—to use any other IP-based network, such as LTE.

3.3 ETCS over LTE

Based on the LTE characteristics, which were described in Section 2.5.5 on page 28, the hypothesis was formulated that LTE can become a future alternative to GSM-R. The logical architecture of the LTE network deployed in a railway environment is illustrated in Figure 3.3.

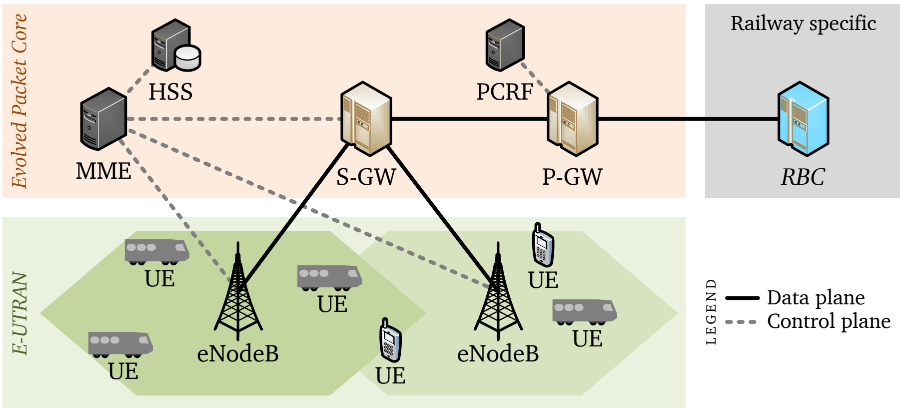


Figure 3.3: LTE architecture in a railway environment

The radio access part of LTE is called Evolved Universal Terrestrial Radio Access Network (E-UTRAN) [38, p. 26]. It consists of User Equipments (UEs) and E-UTRAN NodeBs (eNodeBs). UE is a user terminal, which, in the railway case, can be either a handheld device or a radio module installed on a train. eNodeB is a base station providing radio coverage, managing radio resources, and scheduling packets.

The backbone of the LTE network is called EPC [38, pp. 27–28]. It consists of the following logical nodes:

- *Serving Gateway (S-GW)*, which is responsible for providing the interconnection between Packet Data Network Gateway (P-GW) and eNodeB. This is done using GPRS Tunnelling Protocol (GTP) [25, p. 214].
- *Packet Data Network Gateway (P-GW)*, which is a border node between the LTE and an external network, e.g. a fixed railway communication network. Therefore, the P-GW is the node providing interface to the RBC.
- *Mobility Management Entity (MME)* is responsible for all internal network signalling between eNodeB and the core network. It handles authentication, bearer management, mobility management, and interconnection with other radio networks [25, p. 213].

- *Home Subscriber Register (HSS)* is a database of subscriber (user) information. It includes user profiles, which, for example, list the services available for a particular user and the external networks the user can access [39, p. 34].
- *Policy and Charging Rules Function (PCRF)*, which handles QoS policies and rules, e.g. regarding bearer establishment [39, p. 32].

Depending on the implementation choice, the EPC logical elements may be placed in separate physical nodes or may be collocated [39, p. 26], as in the simulation model that is presented later in this chapter.

Originally, the “LTE” name was used for the evolution of the radio part of the network, i.e. E-UTRAN. The evolution of the backbone part of the network, i.e. EPC, is referred to as System Architecture Evolution (SAE). The whole architecture of the network is called Evolved Packet System (EPS) [38, p. 25]. However, commonly, the whole EPS is referred to as LTE [25, p. 123], and, therefore, this naming approach is used in this thesis.

3.3.1 Protocol stack

As mentioned in section 3.2, the move to LTE would require adaptation of ETCS and Euroradio to the IP-based protocol stack [11, p. 49]. However, the IP-based protocol stack for ETCS is not specified [4, p. 163]. Thus, it is an open question how it should be constructed.

One of the basic choices is what functionality should be included in the new Euroradio layer. One option is to keep this layer responsible for authentication, encryption and, data integrity (retransmissions), in the same way as it is currently in GSM-R. Another option would be to move some of these functionalities to standard industry protocols known from other IP networks. For instance, the data integrity could be provided by the Transmission Control Protocol (TCP), while encryption could be provided by the Internet Protocol Security (IPsec). There are also other possibilities, for example, some researchers propose to use a version of Multipath TCP (MPTCP) in order to improve communication reliability through multipath redundant transmission [56].

In this work, it was assumed that the Euroradio functionality will remain unchanged. Based on this assumption, a new protocol stack for OBU-RBC communication over LTE is proposed, as shown in Figure 3.4.

ETCS datagrams on the way between the OBU (UE) and the RBC pass through the eNodeB and the S-GW/P-GW (in this example, the two EPC gateways are collocated). Figure 3.4 illustrates the protocol stack in each of the nodes.

Looking at the stack from the top, in the two end nodes, the first layer is ETCS (i.e. the application layer). Below, there is the Euroradio layer. Then, the User Datagram Protocol (UDP) is chosen as the transport layer.

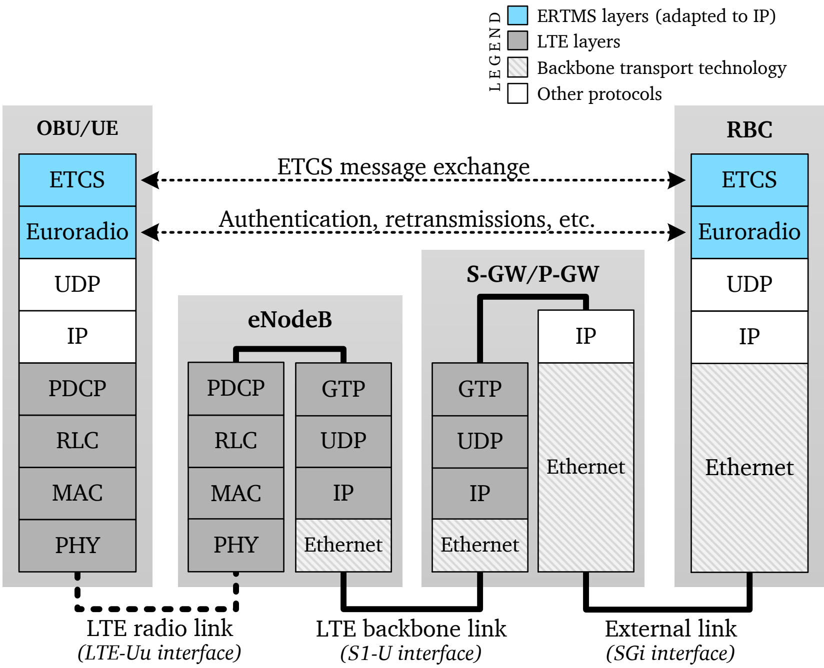


Figure 3.4: Proposed OBU-RBC communication based on the LTE network

An alternative transport protocol was TCP. However, many of the TCP features, such as the slow-start and the congestion control, would be excessive for the infrequent low-rate ETCS traffic. Secondly, TCP would introduce more overhead due to its larger headers compared to UDP. Thirdly, the retransmission mechanisms at the transport layer would be redundant to the similar mechanism provided by Euroradio. Finally, TCP would only allow retransmission of a packet containing the same data as the original lost version. In opposite to this, a retransmission mechanism at the higher layer is able to update the data before each retransmission attempt (e.g. with an updated train position). Therefore, UDP was selected as the transport protocol.

In the lower part of the stack—below the IP layer—there are three types of links (interfaces) and the respective protocol sets:

- *LTE-Uu interface* (radio interface) interconnects the UE and the eNodeB. The following protocols are used on this interface: Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC), Medium Access Control (MAC), and Physical Layer (PHY).
- *S1-U interface* (backbone interface) interconnects the eNodeB with the EPC

gateways (S-GW and P-GW). The protocol stack on this interface consists of: GTP, UDP, IP, and an underlying transport technology. In this work, it was chosen to use Ethernet as the backbone transport technology.

- *SGi interface* provides connectivity to the external network. Ethernet was also chosen here as the transport technology.

3.4 ETCS transmission requirements

As it was explained in Section 3.1, the quality of the OBU-RBC communication directly affects the operation of the ETCS system and, therefore, also the train operation. Timely and reliable message exchange between the OBU and the RBC allows trains to run safely and efficiently (in terms of speed and track capacity utilization). Hence, the railway industry defined a set of requirements that must be fulfilled by the communication network supporting ETCS [19, 22, 30]. These requirements are summarized in Table 3.1. It is important to note that the requirements apply to the end-to-end transmission, i.e. from the OBU to the RBC. Also, the requirements should be fulfilled regardless of the network load [19, p. 17].

The ETCS requirements concern five areas: radio coverage, user speed, transfer delay, communication disruptions, and communication establishment.

Table 3.1: Summary of ETCS transmission requirements, based on [30].

Parameter	Value
Minimum received signal power [22, p. 41]	−92 dBm
Maximum supported user speed [22, p. 27]	500 km/h
Maximum end-to-end delay for a 30-byte data block	≤ 0.5 s
Transmission interference period	< 0.8 s (95%) < 1 s (99%)
Error-free period	> 20 s (95%) > 7 s (99%)
Transmission break during handover [19]	< 0.5 s
Connection establishment delay	< 8.5 s (95%) ≤ 10 s (100%)
Connection establishment error probability	10^{-2}
Connection loss rate	$\leq 10^{-2}$/h
Network registration delay	≤ 30 s (95%) ≤ 35 s (95%) ≤ 40 s (100%)

The *radio coverage* requirement defines the minimum acceptable power of a radio base station signal, i.e. the downlink power. Anywhere within the railway area, the signal power must be above −95 dBm, but the recommended minimum power is −92 dBm [22, p. 41]. Moreover, in order to take into account signal fading, e.g. due to shadowing and multipath propagation, in practice, the radio coverage is planned with a higher power target. For instance, −82 dBm target is used in the Swedish GSM-R network [57, p. 16], while −77 dBm target is used in the Norwegian network [58, p. 17].

The second requirement—on the *user speed*—defines that the network should support user terminals travelling at the speed of up to 500 km/h.

Transfer delay requirement defines the maximum time that it may take to deliver a 30-byte data block over the network. According to [30, p. 15], it must be below 0.5 s (in 99% of cases).

Another group of the requirements concerns *communication disruptions*, which are defined in terms of the maximum interference period, the minimum error-free period following the interference, and the maximum transmission break due to an inter-cell handover.

The last group of requirements concerns *communication establishment* in terms of the maximum network registration time and the maximum connection establishment time. These parameters may affect train operation especially in one scenario, namely, when a running train is about to enter an ETCS area. In such a scenario, communication with the RBC must be established before the train approaches the transition point, i.e. the point of switch from the light-based signalling to the ETCS signalling. Therefore, network registration and connection establishment must be fast enough, otherwise, train will be forced to stop [53, pp. 32–39].

3.4.1 Requirements for packet-switched transmission

In GSM-R, ETCS messages are transported via Circuit-Switched Data (CSD) calls. Accordingly, ETCS transmission requirements has been defined under the assumption that the supporting mobile network is circuit-switched based [30, p. 12]. If GSM-R is to be replaced with a packet-switched technology, such as LTE, then these transmission requirements must be redefined. Work on this redefinition has been initiated, but has not been finalized.

However, Banedanmark—the Danish rail infrastructure manager—published tentative requirements for packet-switched transmission of ETCS messages [19, p. 18]. This tentative requirements, concern mainly two areas: the transfer delay and the data integrity, as listed in Table 3.2. In the following research work, these tentative requirements were used to evaluate LTE as an alternative to GSM-R.

Table 3.2: Tentative ETCS requirements for packet-switched networks [19]

Parameter	Value
Mean end-to-end delay for a 128-byte packet	< 0.5 s
End-to-end delay for a 128-byte packet (95% cases)	< 1.5 s
Mean end-to-end delay for a 1024-byte packet	< 2.0 s
End-to-end delay for a 1024-byte packet (95% cases)	< 7.0 s
Data loss probability	10^{-4}
Data duplication probability	10^{-5}
Data corruption probability	10^{-6}
Out-of-sequence data delivery probability	10^{-5}

3.5 Factors affecting ETCS transmission in LTE

The goal of the research work presented in this chapter was to analyse transmission performance offered by the LTE network, as presented in Section 3.3, and to compare it with the ETCS requirements, as presented in Section 3.4. The performance offered by the LTE network depends on various factors. The most important ones were identified as follows [Sniady2013c]:

- *eNodeB deployment density*, i.e. the number of cells deployed for providing radio coverage over a railway line.

  The number of eNodeBs is expected to affect, among others: the number of inter-cell handovers, the average radio utilization, and the radio transmission power.

- *Train (UE) speed*

  Train speed is expected to affect, among others: the handover frequency and the signal quality (e.g. due to channel estimation).

- *Traffic load on the communication network*

  Traffic load is expected to affect, among others: the queuing delay, the radio utilization, and the control channel utilization.

- *QoS mechanisms*

  The railway LTE network is expected to provide a range of applications with different importance for railway operation [52]. Therefore QoS mechanisms is essential for ensuring prioritization of the safety-critical applications, such as ETCS.

3.6 ETCS simulations

In order to validate ETCS transmission in LTE, a simulation-based approach was chosen. More specifically, it was decided to use OPNET Modeler v. 17.5.A PL5 [59] due to the following reasons [Sniady2013c]:

- OPNET provides a complete model of the end-to-end LTE network. The model includes the full protocol stack for both the radio and the backbone parts. As shown in Figure 3.5, all protocol layers are included in the model. Thanks to these, in the simulations, it is possible to observe the end-to-end transmission performance. This is very important, since ETCS requirements are defined using end-to-end performance metrics. Therefore, only such a complete model can produce results that can be confronted with the requirements.

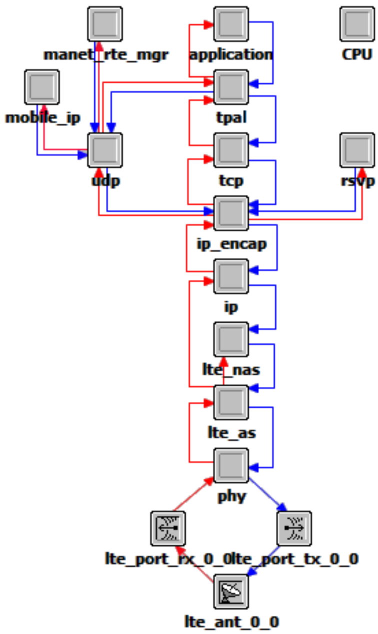


Figure 3.5: OPNET model of the UE node

- Additionally, the OPNET LTE model includes a full set of fundamental LTE mechanisms, such as: QoS provisioning using EPS bearers, Radio Link Adaptation, Dynamic Scheduling, and Mobility Management. Therefore, it is possible to model a realistic network performance.

- OPNET can also model node movements. In many railway scenarios, train movement is an important factor that must be taken into account. Therefore, thanks to the movement modelling in OPNET, it is possible to build scenarios that simulate realistic railway operation.

- Finally, OPNET provides various methods for modelling application traffic. There is a set of built-in models, which generate traffic typical for the commonly used applications, such as video and voice transmission. However, there are also more advanced method, such as AppTransaction Xpert (ATX) Whiteboard, which is an additional tool complementary to OPNET Modeler. ATX allows modelling of a distributed system with multiple tiers. Detailed message flows between the tiers can be then defined, including very specific interactions between particular messages.

3.6.1 ETCS model in OPNET

Figure 3.6 illustrates the basis OPNET model of the ETCS OBU-RBC communication over an LTE network. The setup consists of four nodes: UE, eNodeB, EPC and RBC. Besides the EPC, each node in the model represents a single logical node of the LTE network. Only the EPC node includes functionality of multiple logical elements, namely: the S-GW, the P-GW, and the MME.

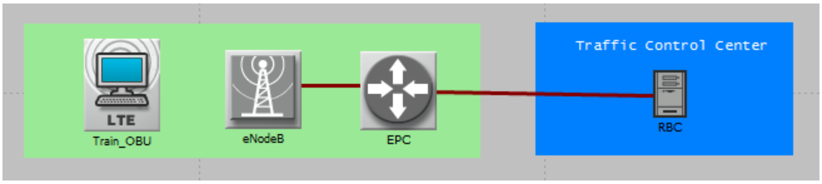


Figure 3.6: Basic OPNET model of the LTE network for ETCS signalling

On top of this network model, an *ETCS application model* has been developed using OPNET ATX. The aim of this model is to generate traffic representing typical ETCS traffic sent through the railway mobile network. Moreover, the model records various statistics—such as the transfer delay and the message loss—that measure transmission performance between the OBU and the RBC. The application model consist of three phases.

The **first application phase** models ETCS session establishment, i.e. the message exchange between the OBU and the RBC during the so called "start of the mission". This ETCS message exchange, which is illustrated in Figure 3.7, is executed when the OBU is started and a session with the RBC must be established. This is a necessary step before the train movement supervision may begin.

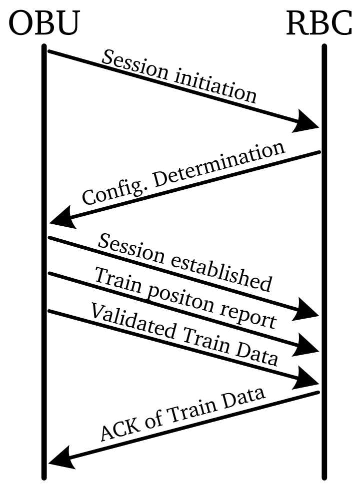


Figure 3.7: ETCS message flow during ETCS session establishment (the first application phase). Detailed source code of an OPNET ATX process modelling this application phase is presented in Appendix A.1.

The **second application phase** models the MA extension procedure, which was described in Section 3.1 on page 34. The ETCS message exchange during this procedure is shown in Figure 3.8. The OBU sends an MA request to the RBC. The request contains an up-to-date train position report. The RBC replies to the request with an updated MA grant, which allows the train to continue driving.

The ETCS message flows—in both phases—are modelled based on an example published by Stanley (ed.) [4, p. 108].

The MA extension procedure is repeated many times as the train is running along a railway line. In a real system, the time interval between MA extensions

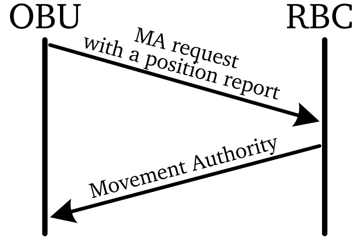


Figure 3.8: ETCS message flow during MA extension procedure (the second application phase). Detailed source code of an OPNET ATX process modelling this application phase is presented in Appendix A.2.

depends on railway infrastructure (e.g. Eurobalise placement), timetables and actual train movements. In the model, this interval is a random value based on the uniform distribution in a configurable range.

The MA extension interval based on random distribution is the first simplification of the model. The second one is a fixed ETCS message length. In a real system, ETCS messages have variable lengths. However, the ETCS requirements are defined for messages with a fixed length. Therefore, the herein presented model also generates fixed-length messages. This allows to compare simulation results directly with the railway requirements.

The **third application phase** is executed at the end of a simulation run. During this phase, no messages are exchanged, but only the final statistics are calculated.

Data integrity protection

All of the ETCS messages sent between the OBU and the RBC are protected by an end-to-end retransmission mechanism, whose responsibility is to ensure ETCS data integrity. Thus, the mechanism models the basic functionality of the Euroradio.

Whenever a sender node transmits a message, an associated timer is started, as is illustrated in Figure 3.9. Reception of every message is expected to be confirmed with a 5-byte acknowledgement message (ACK). If the ACK does not arrive until the timer duration expires, the sender attempts to retransmit the message. The timer duration and the maximum number of attempts are configurable.

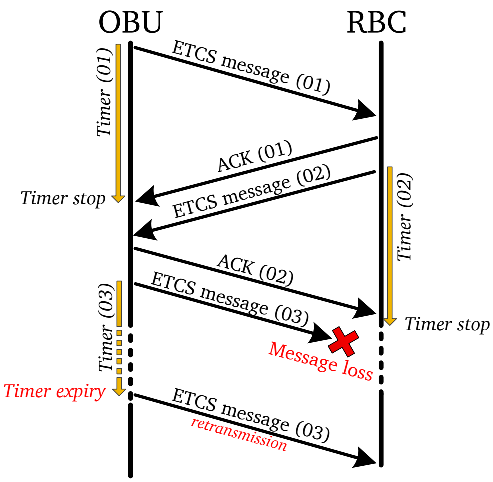


Figure 3.9: Retransmission of a lost ETCS message. Detailed source code of an OPNET ATX process modelling this retransmission mechanism is presented in Appendix A.3 on page 158.

The ACK mechanism was implemented using separate 5-byte messages. An alternative solution was to include an acknowledgement field within the ETCS message header. For instance, in Figure 3.9, Message 02 could carry the acknowledgement for Message 01, and so on. However, the problem with such a solution is a long inactivity interval between the consecutive ETCS messages. Even in case of closely interrelated messages, such as the MA request and the MA grant, there may be a relatively long time before the first one is received and the second one is sent. This is because, after the MA request is received, the MA grant reply must wait, for example, until track switches are set and locked. This mechanical operation may take few seconds. Therefore, without separate ACK messages, the retransmission mechanism would have to operate using very long time-outs. This would significantly increase delay in case of any message loss.

3.7 Impact of the radio deployment on ETCS

While designing an LTE network for railways, the principal parameter that must be chosen is the number of base stations used for providing the necessary radio coverage. It may be decided to deploy a relatively few base stations (eNodeBs), which would transmit at a high power. This setup would cover a railway line with just a few large radio cells. Alternatively, the same railway line may be covered with more base stations, which transmit at a lower power. Thus, the coverage would be provided by many—relatively small—radio cells.

The chosen deployment strategy has an impact on the capacity, relative traffic load per cell, interference and handover frequency. Therefore, the deployment may have an impact on the performance of ETCS transmission.

The aim of the work presented in this section was to analyse the LTE radio deployment strategy in terms of eNodeB density and eNodeB transmission power. A range of scenarios with various number of eNodeBs was evaluated regarding their impact on ETCS transfer delay and data integrity. The analysis presented in this section is based on a previously published work [Sniady2014b].

3.7.1 Radio coverage planning

A railway mobile communication network must provide sufficient coverage, in terms of the received signal power, over the entire railway area where the ETCS-equipped trains operate.

According to the ETCS radio coverage requirement (see Table 3.1 on page 40), the minimum acceptable power of the received downlink signal is −92 dBm [22, p. 41]. The same minimum power requirement ($P_{min} = -92\,dBm$) was applied in the analysis presented herein.

However, it should be noted that the receiver sensitivity in LTE depends on many factors, such as bandwidth, modulation, and receiver noise figure. For example, assuming a 5 MHz bandwidth, −92 dBm power is sufficient to receive a signal with 16QAM modulation and 1/2 channel coding. In case of a more robust modulation, e.g. QPSK, even a signal with a received power of −100 dBm is usable—at the expense of achievable throughput [38, p. 479]. Therefore, in the LTE network, even if the −92 dBm target would not be met, the connectivity should still be available.

LTE cell range as a function of eNodeB transmission power

For the purpose of the radio coverage planning, the relation between the transmission power and the cell range must be found.

P_t is the power on the output of eNodeB transmitter. As the radio signal propagates through space, it is attenuated due to various physical phenomena, such as free space loss, reflection, etc. This effect is called signal path loss. Due to the path loss, the further the receiver is from the transmitter, the lower is the received signal power. Hence, a radio cell has a limited range where the required signal level is maintained.

For the purpose of coverage planning, it is required to find the received eNodeB signal power (P_r) as a function of the transmission power (P_t) and the distance from the eNodeB (d). This will allow the cell range to be estimated as the distance (d_r) where the received signal (P_r) approaches $P_{min} = -92\,dBm$.

Apart from the path loss, other important factors affecting the signal reception must be taken into account, such as antenna gains, cable losses, and interference margin [39, p. 225]. Therefore, on the dBm scale, the received power (P_r) is a sum of multiple contributions, as expressed by the following formula:

$$P_r[dBm] = P_t + G_{enb} - L_{enb} + G_{ue} - L_{ue} - M - L \tag{3.1}$$

where:

- P_t is the eNodeB transmission power,
- G_{enb} is the antenna gain of the eNodeB transmitter,
- L_{enb} is the feeder cable loss at the transmitter,
- G_{ue} is the antenna gain of the UE receiver,
- L_{ue} is the sum of power losses at the receiver, e.g. due to penetration loss,
- M is the margin for interference and fading,
- L is the signal path loss, which can be found as described below.

Signal **path loss** can be estimated using various propagation models. In this work, the modified *COST231 Hata* model is used, as defined in the 3GPP standard [60]. The *COST231 Hata* model includes various cases depending on the environment (urban, suburban). Since an exemplary line that is presented later in this section runs mainly through rural or suburban areas, the *Suburban Macro* path loss model was chosen. In this model, the signal path loss (L) is expressed by the following formula [60]:

$$\begin{aligned} L[dB] = &\left(44.9 - 6.55 \cdot log_{10}(h_{enb})\right) \cdot log_{10}\left(\tfrac{d}{1000}\right) \\ &+ 45.5 + \left(35.46 - 1.1 \cdot h_{ue}\right) \cdot log_{10}(f_c) \\ &- 13.82 \cdot log_{10}(h_{ue}) + 0.7 \cdot h_{ue} + C \end{aligned} \tag{3.2}$$

where d is the distance between the eNodeB and the UE (in m), h_{enb} is the height of eNodeB antenna (m), h_{ue} is the height of UE antenna (m), f_c is the carrier frequency (MHz), and C is a constant factor equal to 3 dB in the urban environment and 0 dB in the suburban.

Cell range is estimated as the distance d_r, where $P_r = P_{min}$. In order to find d_r, the path loss L in Eq. 3.1 is substituted with Eq. 3.2, as expressed in the following:

$$\begin{aligned} P_{min}[dBm] = &P_t + G_{enb} - L_{enb} + G_{ue} - L_{ue} - M - L \\ = &P_t + G_{enb} - L_{enb} + G_{ue} - L_{ue} - M \\ &- \left(44.9 - 6.55 \cdot log_{10}(h_{enb})\right) \cdot log_{10}\left(\tfrac{d}{1000}\right) \\ &- 45.5 - \left(35.46 - 1.1 \cdot h_{ue}\right) \cdot log_{10}(f_c) \\ &+ 13.82 \cdot log_{10}(h_{ue}) - 0.7 \cdot h_{ue} - C \end{aligned} \tag{3.3}$$

By transforming the above equation, the cell range d_r is found as:

$$d_r[m] = 10^{\left(3 + \frac{x}{44.9 - 6.55 \cdot log_{10}(h_{enb})}\right)} \tag{3.4}$$

where:

$$\begin{aligned} x = &P_t + G_{enb} - L_{enb} + G_{ue} - L_{ue} - M - P_{min} \\ &- 45.5 - \left(35.46 - 1.1 \cdot h_{ue}\right) \cdot log_{10}(f_c) \\ &+ 13.82 \cdot log_{10}(h_{ue}) - 0.7 \cdot h_{ue} - C \end{aligned} \tag{3.5}$$

This equation is the basic tool that was used for planning the radio coverage in the following section.

3.7.2 LTE coverage along Snoghøj-Odense line

The above radio coverage considerations were applied to a problem of planning an LTE radio deployment along an exemplary railway line. For this purpose, Snoghøj-Odense line, whose overview is shown in Figure 3.10, was chosen. The line was selected as the best example of a Danish line with a high intensity of a high-speed train traffic.

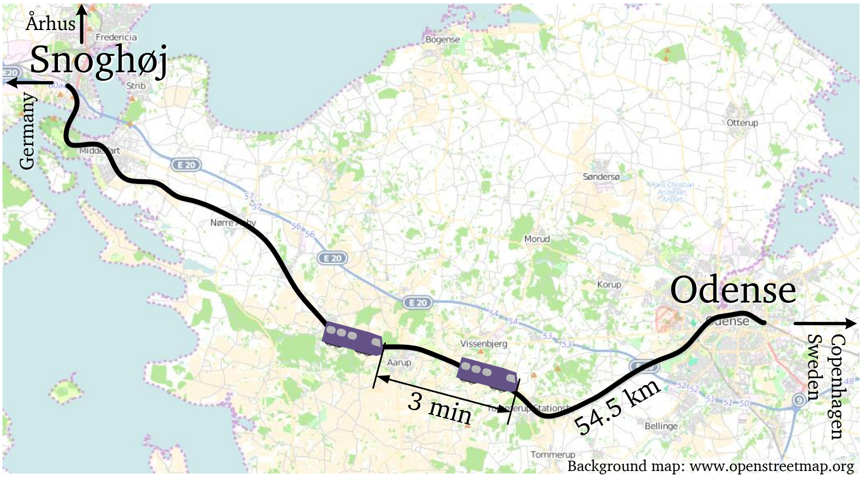


Figure 3.10: Overview map of the Snoghøj-Odense railway line

Snoghøj-Odense is one of the most important lines in Denmark. It is the only one connecting the West and the East parts of the country. Due to that, over 50% of the total train passenger traffic and over 80% of the total train cargo traffic pass over the line [61, p. 5]. It is a typical mainline, which is used primarily by long distance trains. Most of these do not stop on the intermediate stations and pass uninterrupted over the whole railway line. Besides the national trains, there is also intensive international railway traffic. Most of the foreign trains that run through Denmark run through this particular line, because it is the main railway link between Sweden and Germany.

Furthermore, Snoghøj-Odense is one of the lines with a maximum running speed of 180 km/h. Currently, this is the highest speed limit on any railway line in Denmark [20, p. 20]. The line is double track and—in the peak hour—it is used by up to 15 trains in each direction [20, p. 20].

The line is 54.5 km long. Therefore, if a train runs at the maximum speed of 180 km/h, it passes the line in approximately $\frac{54.5\,km}{180\,km/h} \approx 18\,min$. The minimum headway time between trains is 3 min [20, p. 20]. Hence, at any given moment in time, there may be $\frac{18\,min}{3\,min} = 6$ trains running on the line in each direction, i.e.

12 trains in total. In order to take into account trains that may be standing at the intermediate stations, 15 trains are considered in the following analysis.

Number of base stations

Due to the length of Snoghøj-Odensee line, in order to cover it with an LTE network, multiple cells are needed. The number of deployed cells (eNodeBs) is denoted here as N and it is considered in a range between 10 and 55.

The lower boundary of 10 eNodeBs was chosen, because this is the number of GSM-R base stations currently deployed along the line [62] [63, Kort FY1]. In such a setup, each cell must have an approximate radius of 2.5 km. There are approximately $\frac{15\,trains}{11\,eNodeBs} \approx 1.4$ trains per cell.

At the upper limit of 55 eNodeBs, the approximate cell radius is 500 m. There is approximately a $\frac{15\,trains}{55\,eNodeBs} \approx 0.3$ train per cell. Considering this low number of trains and the installation issues, such as equipment costs, mast construction, etc., it is unlikely that more than 55 base stations would be deployed.

Radio bandwidth

In Europe, GSM-R operates in a 4 MHz or a 7 MHz bandwidth [4, p. 148]. LTE, on the other hand, supports only the following bandwidths: 1.4 MHz, 3 MHz, 5 MHz, 10 MHz, 15 MHz and 20 MHz [39, p. 241]. Thus, LTE does not offer a bandwidth that can exactly match the available 4 or 7 MHz.

Unless spectrum allocation for railways is increased in the future, this mismatch between the bandwidths may be an issue. A railway LTE network would have to use 3 MHz out of 4 MHz available, or 5 MHz out of 7 MHz available. However, the remaining 1 or 2 MHz that could not be utilized by LTE could be used to keep a GSM-R network for backwards compatibility. Hence, there are possible solution that would prevent spectrum wastage. Nevertheless, since the spectrum allocation is mainly a political and business issue, it was not considered further in this thesis.

In all scenarios presented in this thesis, the 5 MHz LTE bandwidth was chosen, because it is the closest one to the GSM-R bandwidth.

Carrier frequency

In the analysis, two carrier frequencies are considered: 921 MHz, which is currently used by GSM-R, and 2110 MHz, which is one of the LTE bands commonly used in Europe by the commercial mobile networks.

Other parameters

Other assumptions and parameters used for modelling of the LTE coverage are summarized in Table 3.3. Specific values were chosen based on: the UIC specifications and guides [22, 26], typical values for LTE networks [39, p. 225], and current infrastructure used for GSM-R in Denmark [64]. The chosen values were picked assuming the worst-case conditions (e.g. interference) and hardware (e.g. losses at the transceivers). Hence, the results obtained in the following analysis would be significantly improved, if the scenario assumed more favourable parameters.

Table 3.3: Parameters and assumptions used in the analysis and the following simulations

Parameter		Value
Minimum received power requirement	(P_{min})	−92 dBm
Carrier frequency	(f_c)	921 MHz *or* 2110 MHz
Bandwidth		5 MHz
eNodeB antenna gain[1]	(G_{enb})	15 dBi
Power loss at eNodeB[1]	(L_{enb})	−9 dB
eNodeB height[2]	(h_{enb})	45 m
UE antenna gain[1]	(G_{ue})	0 dBi
Power loss at UE[1]	(L_{ue})	−2 dB
UE height[3]	(h_{ue})	4 m
Interference and fading margin[4]	(M)	8 dB
Constant factor in COST231 Hata model[5]	(C)	0 dB
Handover type[6]		over X2 interface
Channel model[7]		ITU Vehicular A
Railway line length[8]		54.5 km

[1] Following the typical values as published by Neele and Wootton in UIC "GSM-R Procurement Guide" [26, pp. 111–112].

[2] Chosen in accordance with information released by Banedanmark [64].

[3] Assuming that the UE antenna is placed on a train roof [22, p. 41].

[4] The worst-case value selected within a typical range for LTE [39, p. 225].

[5] Assuming a suburban radio propagation model [60].

[6] During a handover in an LTE network, the involved eNodeBs may communicate directly over X2 interface, or through S1 interface—with a help of MME. In the simulations, the X2-based handover was enabled, since it is the more efficient one [25, pp. 245–249].

[7] ITU Vehicular A multipath channel model was chosen due to the mobility of UEs (trains) in the considered scenario. This follows channel choice made by Dusza et al. [65].

[8] According to the report prepared for the Danish Signalling Programme [20, p. 20].

Path loss

By applying the chosen parameters to Eq. 3.2, the relation between the path loss (L) and the distance (d) was found, as shown in Figure 3.11. As expected, the longer is the distance between the transmitting eNodeB and the receiving UE, the higher is the signal path loss. Due to that, the received signal power is lower as the UE moves away from the eNodeB. Also, it is visible that the path loss is significantly higher when the higher carrier frequency (f_c) is used.

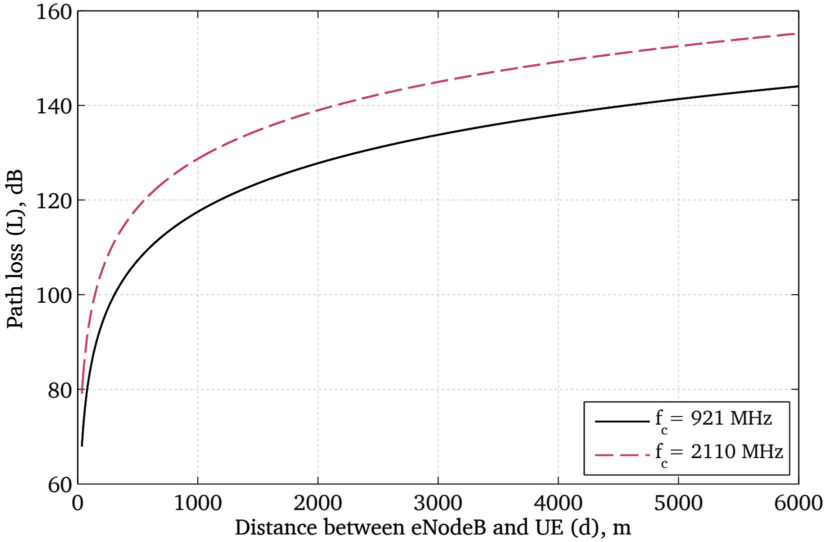


Figure 3.11: Signal path loss in relation to the distance from the eNodeB

LTE cell radius

Regardless of the number of eNodeBs (N) that will be deployed along Snoghøj-Odense railway, the proper radio coverage must be provided over the entire line. Therefore, the LTE cell radius (d_r) has to be adjusted according to the chosen number of eNodeBs (N). This adjustment can be done by selecting the appropriate maximum transmission power (P_t)—following the relation between P_t and d_r defined by Eq. 3.4 on page 49.

Figure 3.12 shows the LTE cell radius (d_r) depending on the eNodeB transmission power (P_t). This relation was calculated specifically for the Snoghøj-Odense example using the parameters from Table 3.3. As expected, the higher is the transmission power, the larger is the radius of a cell. For instance, in case of 921 MHz carrier frequency, if the eNodeB is transmitting at 48 dBm, then the cell radius is

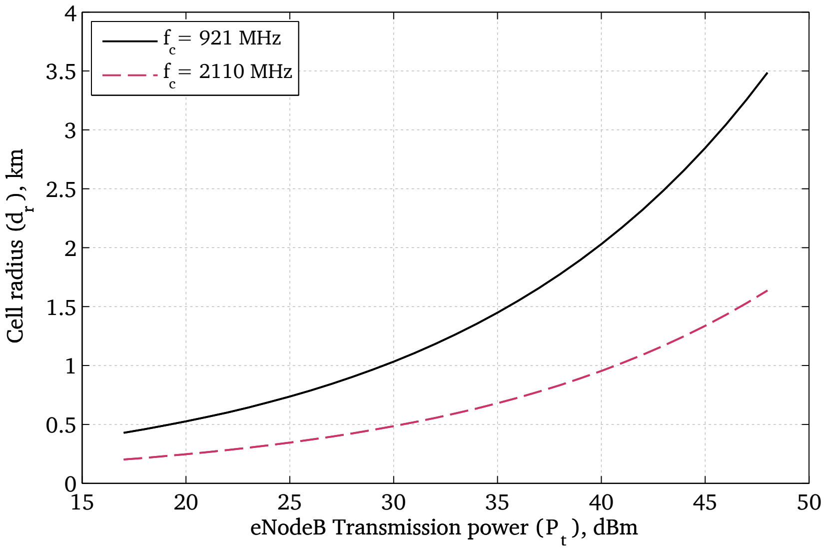


Figure 3.12: Cell range in relation to the eNodeB transmission power

approximately 3.6 km. Since a typical eNodeB antenna is able to transmit with power up to 48 dBm [39, p. 225], higher transmission powers were not considered.

eNodeB transmission power as a function of the deployment density

Depending on the number of eNodeBs (N), each radio cell needs to have its radius (d_r) adjusted. Considering a linear coverage, such as in the Snoghøj-Odense example, the required cell radius can be expressed by the following formula:

$$d_r = \frac{1}{2} \cdot \frac{LineLenght}{N} \tag{3.6}$$

By comparing d_r in Eq. 3.6 and Eq. 3.4 on page 49, it is possible to find the relation between the number of deployed eNodeBs (N) and the eNodeB transmission power (P_t). This relation—for the example of this specific railway line—is illustrated in Figure 3.13. It can be seen that the more eNodeBs are deployed along the line, the lower transmission power per each eNodeB is required. This is due to the shorter radius of each cell. However, on the other hand, more of those eNodeBs must be deployed. Thus, in order to compare the deployments, the total transmission power from all eNodeBs (P_{total}) must be found. On the linear scale, it may be expressed by the following formula:

$$P_{total}[W] = N \cdot P_t[W] \tag{3.7}$$

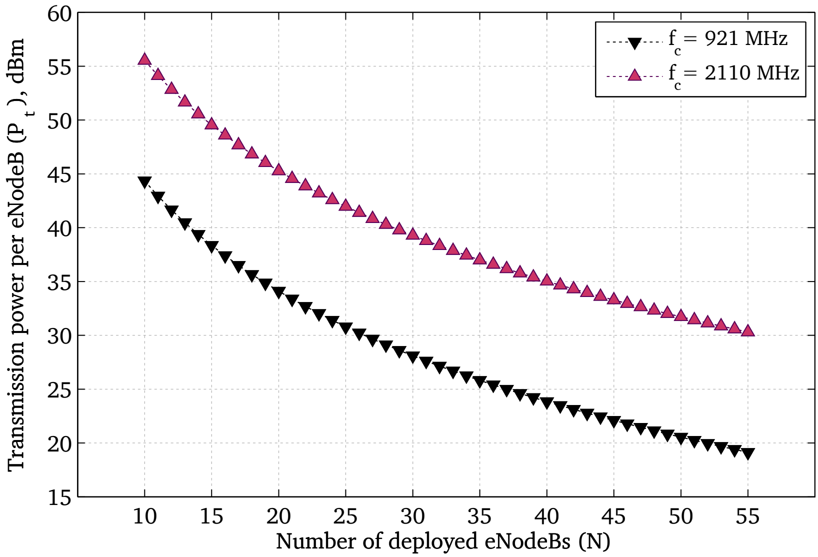


Figure 3.13: eNodeB transmission power in relation to the number of eNodeBs along the line

P_{total} in relation to N is shown in Figure 3.14. The more eNodeBs are deployed, the lower is the total power required to provide the appropriate coverage. Therefore, from the point of view of minimizing the transmission power, it is preferable to deploy as many eNodeBs as possible along the line. This conclusion must be taken into account while evaluating the long-term costs of deploying the railway mobile network.

3.7.3 Simulation model

From the theoretical analysis presented in the previous section, it is possible to draw conclusions on the impact of the deployment density on the required transmission power. A denser radio network uses a lower transmission power. This may be an important aspect to be considered while choosing a radio deployment.

However, another crucial aspect that must be investigated is the impact of the deployment density on ETCS performance in terms of delay and data loss. This is of the highest importance for railways, due to the fundamental role of ETCS in railway operation.

For the purpose of evaluating the impact of the deployment density on ETCS, a simulation scenario was prepared. It modelled an LTE network deployed along the Snoghøj-Odense line. The scenario was considered in 10 cases, which differed

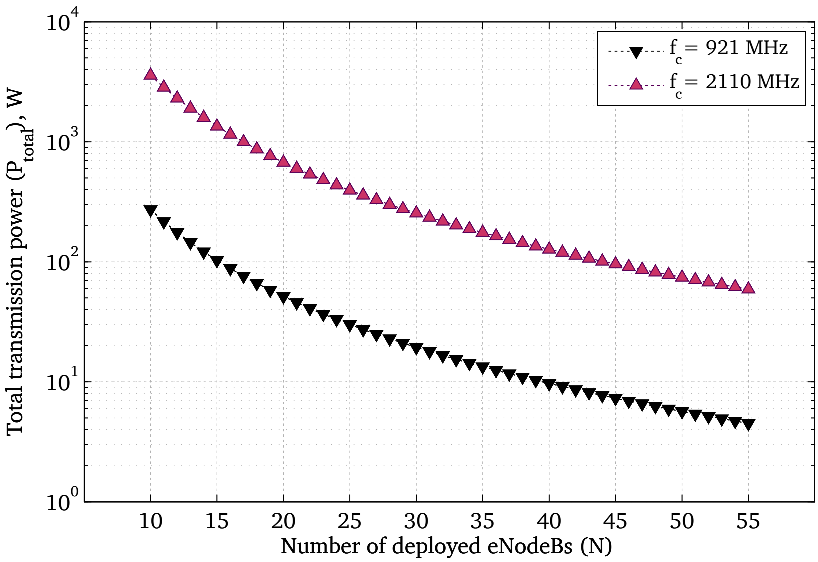


Figure 3.14: Total transmission power of all eNodeBs in relation to the number of eNodeBs along the line. Note the logarithmic scale on the vertical axis.

in the number of deployed eNodeBs. The investigated range was from 10 to 55, as in the theoretical analysis. With the exception of the eNodeB deployment density, the cases were identical. Figure 3.15 shows the first case with 10 eNodeBs.

Following the description from Section 3.6.1 on page 44, the simulation model consisted of an RBC server, an EPC node, two Ethernet switches, multiple eNodeBs and 15 UEs. Each UE was representing a train running at 180 km/h along the line.

The network operated in the 5 MHz bandwidth at 921 MHz carrier frequency, which is used by GSM-R. The eNodeB maximum transmission power was configured for each scenario case individually, following the results shown in Figure 3.13. Other details on the simulation scenario are presented in Appendix B on page 183.

Application mixes

Two different application mixes were considered:

1. **Only ETCS:** In the first mix, ETCS was the only application provided by the network. It was assumed that the railway line is divided into 1.5 km long blocks (i.e. train detection sections). This is a typical block length used in railways [53, p. 49]. While approaching the next block, train OBU sends an ETCS position update to the RBC. Assuming the 1.5 km block length and

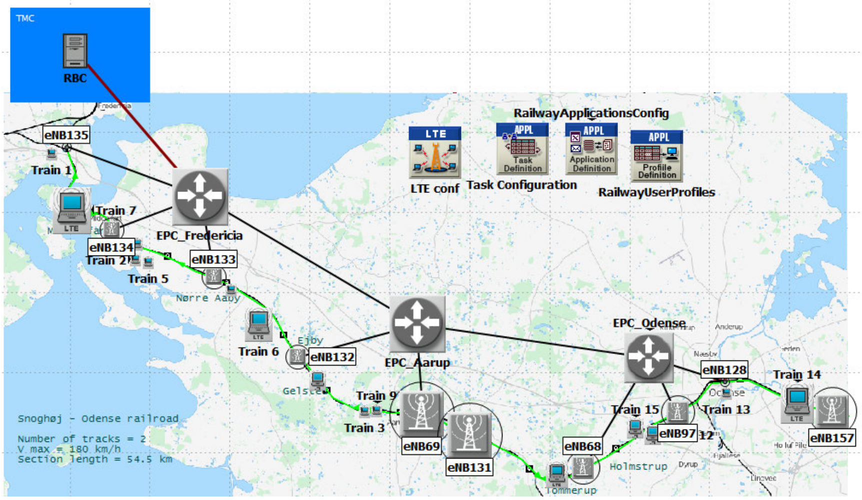


Figure 3.15: Model of the LTE network deployed along Snoghøj-Odense railway line. The case with 10 eNodeBs is shown.

the 180 km/h train speed, the time interval between the position reports is approximately $\frac{1.5\,km}{180\,km/h} = 30\,s$. Therefore, the ETCS application—in each train—was configured to send a position report every 30 s, on average. After receiving a position report, the RBC was sending an updated MA to a train. All ETCS messages had constant length of 128 bytes, following the length specified in ETCS requirements [19, p. 18]. The ETCS retransmission timer was set to 500 ms.

2. **Full application mix:** In the second mix, besides ETCS, a full set of background application was added. For each UE (train), the mix consisted of the following applications: voice communication, tele-maintenance, passenger information, and video surveillance. In order to support these new applications, additional servers were introduced in the network, besides the RBC. The details of this mix are presented later in Section 3.9.2 on page 72.

   For the purpose of the following simulations, the configuration of the background applications was not important, except from the fact that these applications were constantly generating uplink and downlink data traffic.

3.7.4 Simulation results

Each simulation run lasted for 18 minutes, in order for a train, which travelled at 180 km/h, to pass the entire line. Every run was repeated at least 30 times with varying seed numbers.

This section presents ETCS transfer delay and ETCS data loss results collected in these simulations. These two metrics were chosen, because they are the most important for ETCS operation and they are the most likely to be affected by the radio deployment density.

ETCS transfer delay

Figure 3.16 shows the mean ETCS transfer delay in relation to the number of eNodeBs deployed along the railway line. The delay was measured between the OBU and the RBC. Two data series are plotted: the first one represents results collected with ETCS as the only application in the network, while the second series was collected with the full application mix.

Looking at the first series, representing the *ETCS-only mix*, the mean ETCS transfer delay was between 16 ms (the case with 10 eNodeBs) and 19 ms (the case with 55 eNodeBs). Thus, the more eNodeBs were deployed, the slightly longer was the transfer delay.

Looking at the second series of results, collected with the *full application mix*, the mean ETCS transfer delay was between 17 ms (the case with 10 eNodeBs) and

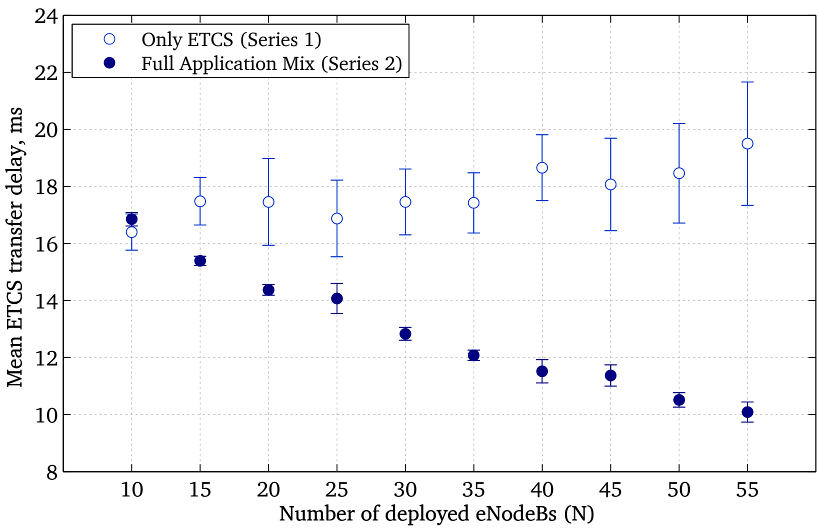


Figure 3.16: Mean ETCS transfer delay in relation to the number of eNodeBs deployed along the line (N). Two simulation series were considered: (1) only with ETCS traffic, and (2) with a full application mix including streaming traffic. Error bars indicate 95% confidence intervals.

10 ms (the case with 55 eNodeBs). Hence—in opposite to the first case—the more eNodeBs, the shorter was the transfer delay.

This difference between the two series may be counter-intuitive, since in the scenario with higher traffic load, the delay was lower. However, it is a consequence of the difference in the Radio Resource Control (RRC) states and the consequent differences in mobility management. An UE can be either in RRC-Idle or RRC-Connected state [39, p. 147]:

- In the *RRC-Idle* state, the UE is registered in the network, but it is not transmitting any data. The UE is only monitoring eNodeBs paging channel and performing signal measurements. In this state, the UE is responsible for mobility management, namely for cell re-selection. Thus, the network does not know the precise location of the UE. Moreover, the UE is not synchronized with any eNodeB.

  If a new data is to be sent or received by the UE, first, the UE must switch to the RRC-Connected state. This involves random access procedure, which allows the UE to synchronize with the selected eNodeB. In case the new data is sent in uplink, the random access procedure is initiated directly by the UE. In case of downlink data, the random access procedure is initiated by the UE in response to a paging procedure. The purpose of the paging is to determine in which cell the UE is located (this is necessary since the network does not know the exact UE location) [39, p. 84]. Due to these time-consuming procedures, there is an initial delay in case an UE being in RRC-Idle state has to receive or send data.

- In the *RRC-Connected* state, the UE is synchronized with the eNodeB and the data transmission is ongoing. In this state, it is the network side, namely eNodeBs, that handles UE mobility. If the UE moves from one cell to another, the relevant eNodeB initiates an inter-cell handover procedure. Therefore, as long as the UE is in the RRC-Connected state, the network knows its exact location, i.e. the exact cell where the UE is at the moment.

In case of the first application mix, the UEs were transmitting and receiving *only ETCS traffic*. An ETCS message was transmitted once in 30 s, on average. During the transmission, the UE remained in the RRC-Connected state. However, once the message was successfully sent (or received), the UE switched to the RRC-Idle state after a 10 s inactivity period. Since the average time between ETCS transmissions was longer than the inactivity period, most of ETCS messages were arriving at the radio interface when the UE was in the RRC-Idle state. Thus, the switch to the RRC-Connected state had to be made before the message could be transmitted. As explained, this state switch involves the random access and the paging procedures, thus, it was a major contributor to the ETCS transfer delay.

In case of the *full application mix*, the traffic load on the network was significantly higher and every UE was continuously transmitting some data (e.g. a video stream from an on-board surveillance camera). Therefore, even though ETCS transmission rate was the same, all UEs remained constantly in the RRC-Connected state due to the background traffic. ETCS message transmission did not involve the time-consuming random access and paging procedures. As a consequence, the mean delay was lower than in the ETCS-only case.

However, in the case with the *full application mix*, the traffic load due to the background applications had also negative consequences. Looking again at Figure 3.16, it is visible that the more eNodeBs were deployed, the lower was the mean ETCS transfer delay. In order to explain this trend, the utilization of the radio channel must be analysed. Figure 3.17 shows the mean utilization of the physical LTE channels in relation to the number of deployed eNodeBs. The results are presented separately for the Physical Uplink Shared CHannel (PUSCH) and the Physical Downlink Shared CHannel (PDSCH).

In all cases—regardless of the number of eNodeBs along the line—the total traffic load on the network was the same. However, this traffic was distributed over a different number of eNodeBs depending on the deployment choice. The

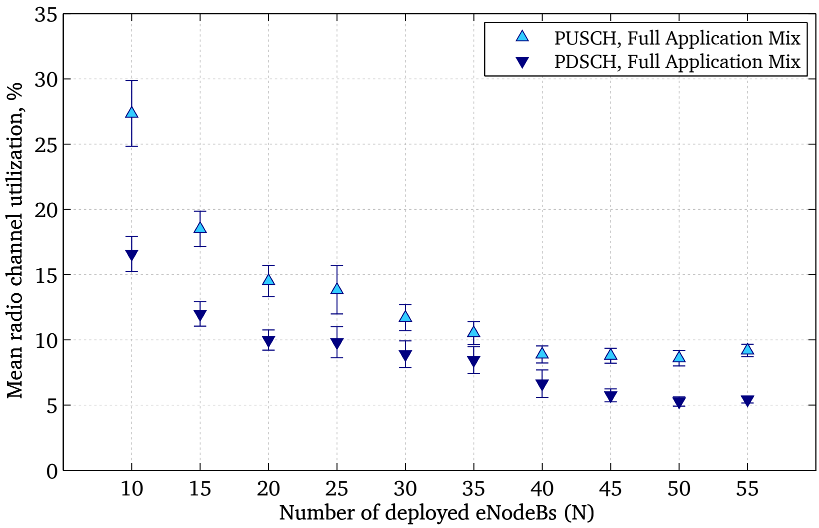


Figure 3.17: Mean physical radio channel utilization in relation to the number of eNodeBs deployed along the line (N). The utilization was measured only at the periods when at least one UE was in the cell. Error bars indicate 95% confidence intervals.

more eNodeBs were deployed, the lower was the average traffic load per cell—consequently, the lower was the mean radio channel utilization.

In a densely deployed network, there were many more eNodeBs than UEs. For example, in the case with 55 eNodeBs, there were only $\frac{15}{55} \approx 0.3$ UE per cell. Therefore, on average, not only there was just a single UE in a cell, but also the two neighbouring cells were empty. Therefore, the inter-cell interference and the packet queuing on the radio link were minimal.

On the other hand, in a sparsely deployed network—with fewer eNodeBs—there were more UEs per cell. For example, in the case with 10 eNodeBs, there were $\frac{15}{10} = 1.5$ UE per cell, on average. Due to that, the radio utilization was relatively higher. Consequently, the inter-cell interference was higher. Due to that interference, the radio transmission error rate was higher and the radio throughout was lower [66]. In turn, this increased the mean ETCS transfer delay, as was shown in Figure 3.16.

As presented earlier in Table 3.2 on page 42, the mean ETCS transfer delay must not exceed 500 ms. The mean transfer delay results observed in the simulations did not exceed 25 ms. Therefore, they were over 20 times shorter than the maximum acceptable 500 ms.

Another requirement is that 95% of ETCS messages must be delivered within 1.5 s (see Table 3.2). The maximum recorded ETCS delay was 540 ms, thus, 100% of ETCS messages were delivered within the 1.5 s limit. What is more, this maximum delay was caused by the large value of the retransmission timer (500 ms). Thus, the maximum delay could be even lower, if the retransmission timer was optimized.

The simulations identified a number of elements that contribute to the ETCS delay, namely: the random access procedure, the paging procedure, and the retransmissions caused by the radio errors. The random access and the paging procedures could be optimized specifically for ETCS, for instance, by changing the inactivity timer duration or by keeping the UEs constantly in the RRC-Connected state. Nevertheless, even without any ETCS-specific optimizations, the modelled LTE network offered delay performance that is significantly better than required by ETCS. This remained true in all of the analysed cases.

ETCS data loss

According to the ETCS data integrity requirements, the maximum acceptable ETCS data loss probability is 10^{-4} (see Table 3.2 on page 42). In the simulations, no data loss was observed, thanks to the radio and the end-to-end retransmission mechanisms. Therefore, the ETCS data integrity requirement was fulfilled.

Nevertheless, packet loss on the radio link was observed. In order to prevent consequent ETCS data loss, the end-to-end retransmissions had to be used (see the data integrity protection mechanism described in Section 3.6.1 on page 46). The retransmission rate was approximately 5×10^{-5}. Thus, one out of 20 000 ETCS messages had to be retransmitted. This low retransmission rate means that even without this protecting mechanism, the requirement on ETCS data integrity would still be fulfilled.

ETCS data integrity in LTE and mechanism for minimizing the data loss probability are investigated in detail in Section 5.4 on page 132.

3.8 Impact of the train speed on ETCS

Currently, the Snoghøj-Odense railway line allows trains to run at a speed up to 180 km/h [20, p. 20]. This was the assumption used in the simulations described in the previous section. However, the railway mobile communication network is required to support trains travelling with a speed up to 500 km/h [30, p. 13, Sec. 6.3.1.4]. Therefore, the aim of the work presented in this section was to:

- Identify the limitations of LTE in a high-speed environment.
- Investigate—using the Snoghøj-Odense scenario presented previously—how does the train speed impact ETCS transmission performance in an LTE network.

The analysis presented in this section is based on a previously published research work [Sniady2013a].

3.8.1 LTE in a high-speed environment

According to the requirements established by the 3GPP LTE is supposed to support user speed up to 500 km/h [67, Sec. 7.3]. Despite that, the LTE transmission in a high-speed scenario is worse compared to static or slow-speed scenarios. This is caused by a number of factors, the most important being:

- **Inter-Carrier Interference (ICI):** OFDM, which is used in LTE radio transmission, divides the frequency spectrum into narrow 15 kHz carriers. In a high-speed scenario, due to the Doppler shift, the orthogonality between these carriers may be broken. This leads to the ICI, which causes then an increase in radio error rate [38, p. 134]. Consequently, the radio throughput is reduced [68].

  Martín-Vega et al. [68] demonstrated that the ICI issue can be successfully minimized using MIMO techniques. Feng et al. [48] proposed an effective ICI cancellation method, which is based on an improved channel estimation. Both solutions offer good performance even up to speed of 500 km/h.

- **Inter-Symbol Interference (ISI):** In OFDM—besides ICI—there is also a possibility of interference between the consecutive symbols, namely the ISI. It appears when the Cyclic Prefix (CP) preceding an OFDM symbol is shorter than the delay spread [38, pp. 135–136]. Therefore, the ISI can be addressed in LTE by using the extended-length CP [45].

- **Physical Random Access CHannel (PRACH):** Due to the Doppler frequency spread, Zadoff-Chu (ZC) sequences used in the LTE random access preamble begin to be distorted at speed above 200 km/h. Consequently, the probability of random access failure increases. This issue may be minimized in LTE by "cyclic shift restriction" method, which allows the network to support users at much higher speeds [38, pp. 391–394].

  Furthermore, in order to improve the success rate of the random access procedure, Wu et al. [49] proposed a modified method for generating ZC sequences. Their analytical and simulation results indicate that the random access failures is greatly reduced—even at the speed of 500 km/h.

- **Inter-cell handovers:** The higher the user speed, the faster is takes to travel across a radio cell. Thus, the handover rate is higher. Moreover, while moving from one cell to another (from source eNodeB to target eNodeB), there is a certain overlap area where both of the cells provide good coverage. The full handover procedure—including measurement period, Time-To-Trigger (TTT), and the handover itself—must be completed when the UE is within the overlap area. As the speed increases, the time available for the handover shortens. If the handover procedure is too slow, then there is a risk of a radio link failure [47].

  There are several research works published on LTE handover in high-speed railway scenarios. Dimou et al. [69] demonstrated that the handover failure rate depends also on the number of users in a cell. The reason is that the probability of an error or a delay in LTE control signalling is increased as the number of users grows. According to the authors, handover failure rate—depending on the speed—is in the range 0.3–1.3% in a cell serving 40 UEs. This risk of a handover failure is the reason for avoiding radio cell boundaries in the ETCS-sensitive areas, as explained further in Section 3.9.1 on page 69.

  Luo et al. [47] proposed an optimized scheme for selecting handover triggers, which provides a high probability of a successful handover even at speed of 450 km/h. Furthermore, Li et al. [70] investigated impact of the power measurement period on the handover success rate. They concluded that, even at 540 km/h, the probability that a handover is triggered correctly can reach 99.8%—in a properly configured LTE network.

All in all, there are multiple factors that degrade LTE transmission performance in a high-speed environment. However, all of them can be addressed with the proposed methods or even with the solutions already available in LTE.

The transmission issues at the high speed are problematic mostly for bandwidth-demanding applications. As demonstrated by Dusza et al. [65], LTE throughput is significantly decreased by the user speed. However, the same authors also prove that—if the lower throughput is acceptable—LTE can support speed even exceeding 500 km/h. At 400 km/h, which is already above the maximum train speed used in Europe, LTE network can offer approximately 2 Mbit/s throughput. This is sufficient to fulfil the ETCS communication demand. Considering the low bitrate of ETCS traffic and the relatively small number of mobile users in railways, LTE should be able to provide ETCS signalling even for high-speed trains.

3.8.2 Simulation model updates

For the purpose of analysing train speed impact on ETCS transmission, the previously presented Snoghøj-Odense simulation scenario was modified in the following way:

- From the previous set of 10 simulation cases, only the two edge cases were kept, namely: the case with 10 eNodeBs and the case with 55 eNodeBs. ETCS was the only application transmitted in the network (i.e. the scenario referred to as *Series 1* in the previous section).
- A set of 9 new simulation cases was developed. Each case considered a different train speed in the range between 25 and 500 km/h. It should be noted, that the Snoghøj-Odense line cannot be upgraded to the speeds above 200 km/h, due to the winding route of the line [61, p. 43]. Thus, in reality, a mobile communication network along the line will not need to support trains faster than 200 km/h. Nevertheless, the aim of the simulations was to investigate transmission performance at high speeds, so no limitations due to the rail design were assumed.
- The train OBU sends an ETCS position report to the RBC after passing over an Eurobalise. Thus, the faster the train goes, the more frequent are the ETCS messages (assuming that the MAs have the same length regardless of the speed). In order to take this into account, the original 30 s time interval between the ETCS messages (t_{ETCS}) was adjusted in each case depending on the train speed, according to the following equation:

$$t_{ETCS} = \frac{block\_lenght}{train\_speed} \tag{3.8}$$

where *block_lenght* is the length of a train detection block, i.e. the distance between Eurobalises. A typical block length of 1.5 km was assumed. The higher was the train speed, the higher was the ETCS traffic load on the network.

- For each modelled case, the length of the simulation runs was equal to the time it took a train to travel along the whole line. Hence, the total number of ETCS messages transmitted in each case was the same (although the message/second rate was different).

3.8.3 Simulation results

This section presents the ETCS transfer delay and the ETCS data loss results collected in the 9 simulation cases with varying train speed.

ETCS transfer delay

Figure 3.18 shows the mean ETCS transfer delay in relation to the train speed. The mean delay remained approximately constant at 16 ms in the range between 25 and 150 km/h. This is the speed range typically used on the Snoghøj-Odense line. Thus, the ETCS delay was not affected as long as the train speed remained in the typical operational range.

However, as train speed exceeded 200 km/h, the ETCS delay began to increase. The delay increase differed between the 10-eNodeB and the 55-eNodeB deployments. In the more sparsely deployed network, the delay increase was small. Even in the worst case, i.e. at 500 km/h speed, it reached only 18 ms—only 2 ms more than in the 25 km/h case.

In the network with 55 eNodeBs, the delay increase was noticeable bigger. At 500 km/h, the delay reached its highest value of nearly 26 ms. This means that the more eNodeBs were deployed along the line, the more sensitive was the ETCS delay to the train speed. This was caused by the following:

- The higher was the train speed, the higher was the ETCS message transmission rate, i.e. more messages per second were sent.
- While moving at 500 km/h through the 55-eNodeB network, a train was spending only 7 s in each cell. In the 10-eNodeB network, this time was 39 s. Thus, the frequency of network procedures related to handover (RRC-Connected) or cell re-selection (RRC-Idle) was significantly higher in the 55-eNodeB deployment. These two network procedures require exchange of network control signalling, radio signal measurements, delay estimation, and possible forwarding of user data between the eNodeB. All of these increase a probability of an error and consequent delay. Thus, the denser was the network, the higher was transmission disturbance due to these procedures.

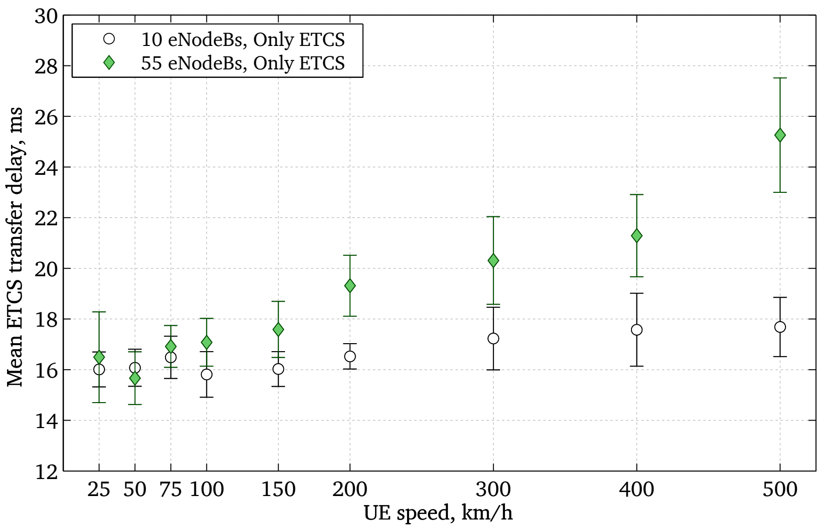


Figure 3.18: Mean ETCS transfer delay in relation to the train speed. Two network deployments were considered: with 10 eNodeBs and with 55 eNodeBs. Error bars indicate 95% confidence intervals.

Nevertheless, the delay increase due to the train speed was not significant enough to exceed the limits set by ETCS requirements. In all of the cases, the mean ETCS delay was at least 20 times smaller than the maximum acceptable 500 ms. Moreover, the average communication interruption due to handover was 13 ms—much lower than the maximum 500 ms allowed by ETCS requirements [19].

ETCS data loss

Thanks to the retransmission mechanism, no ETCS data loss was observed in any of the investigated cases. Thus, the modelled network fulfilled the data loss requirement, which allows a maximum data loss of 10^{-4} [19].

However, similarly as in the previous set of simulations, packet loss was observed on the radio link. Hence, the end-to-end retransmission had to recover the lost ETCS messages in order to prevent the data loss. Figure 3.19 shows the mean ETCS retransmission rate observed in the simulations. By comparing these results with the delay results (Figure 3.18), it is visible that both metrics behaved similarly in relation to the train speed. The reason is that the higher was the train speed, the higher was the probability of disruptions caused by the handovers and cell-reselection.

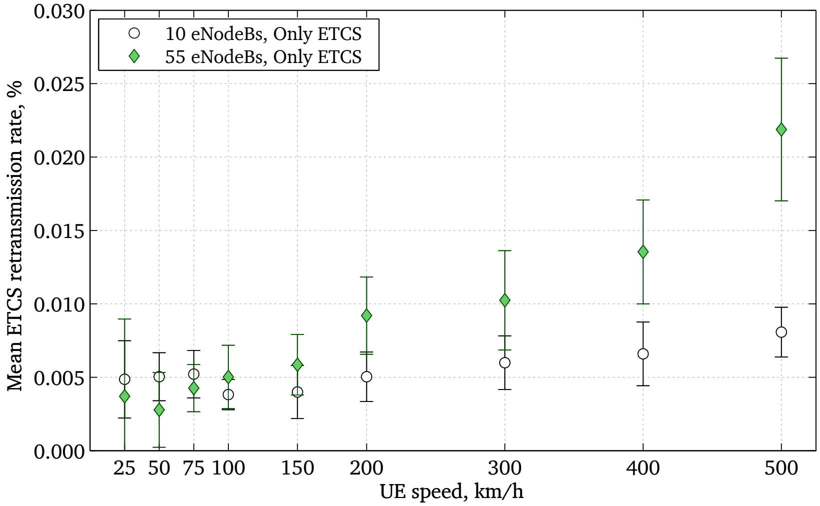


Figure 3.19: ETCS retransmission rate in relation to the train speed. Error bars indicate 95% confidence intervals.

Thanks to the end-to-end retransmissions, the ETCS messages were timely delivered with no data loss. Thus, the modelled network provided ETCS transmission fulfilling the requirements—regardless of the train speed.

All in all, the eNodeB deployment strategy must optimize between the transmission power, the radio capacity and the high-speed performance. The dense base station deployment (with small cells) is advantageous due to the high capacity, lower radio utilization and low power requirements. One the other hand, the sparse deployment (with large cells) is better for handling high-speed trains. This is because, the ETCS performance is less affected by the train speed in such a network.

3.9 Impact of the traffic load on ETCS

The major shortcoming of GSM-R is its insufficient capacity in terms of the number of ETCS sessions handled by a radio cell (see Section 2.4.2 on page 21). This issue is especially problematic in the areas with high-density railway traffic, such as big train stations and junctions. LTE is expected to significantly increase the network capacity and solve the communication bottleneck caused by GSM-R. Accordingly, the goals of the research work presented in this section were to:

- Demonstrate the capacity increase in terms of ETCS sessions per cell (i.e. trains per cell) that can be expected from replacing GSM-R with LTE. The result should explicitly show whether LTE can address the capacity issue that railways currently face.

- Analyse the impact of the traffic load on the ETCS transmission performance in terms of delay and data integrity.
- Determine whether LTE is able to simultaneously provide the critical and the non-critical communication-based applications.

The work presented in this section is based on an example of Copenhagen Central Station, which is an area with the highest train traffic density in Denmark [71, p. 11]. Firstly, the station is analysed regarding the required ETCS capacity. Secondly, new communication-based applications for railways are proposed. Finally, a simulation scenario—modelling the station and the application mix—is used to address the three goals listed above.

This section is based on a research work that was previously published in a paper [Sniady2013d].

3.9.1 Copenhagen Central Station

In this work, Copenhagen Central Station (in Danish: *Københavns Hovedbanegård*) was chosen as the best example for analysing LTE capacity in the railway context. An overview of the station is shown in Figure 3.20.

Copenhagen Central Station consists of: platform tracks and approach tracks at the North and South ends of the station. The station has three entry/exit directions: towards Høje Taastrup, towards Kastrup and towards Helgoland (via an underground tunnel).

Copenhagen Central Station is the biggest train station in Denmark in terms of the number of trains, and also the biggest in terms of the number of served long-distance passengers [71, p. 11]. Moreover, the entry/exit tracks of the station are the busiest sections of the Danish national railway network. Up to 86 arrivals/departures per hour must be handled by the station [20, p. 20].

Due to this high density of train traffic, the station has been identified, by Banedanmark, as the main area of concern regarding the capacity of the Danish GSM-R network [19, p. 3]. It is the place where GSM-R network may not be able to provide sufficient number of communication channels to serve all ETCS-equipped trains. Due to that, it is also the best example for validating the capacity increase that can be offered to railways by LTE.

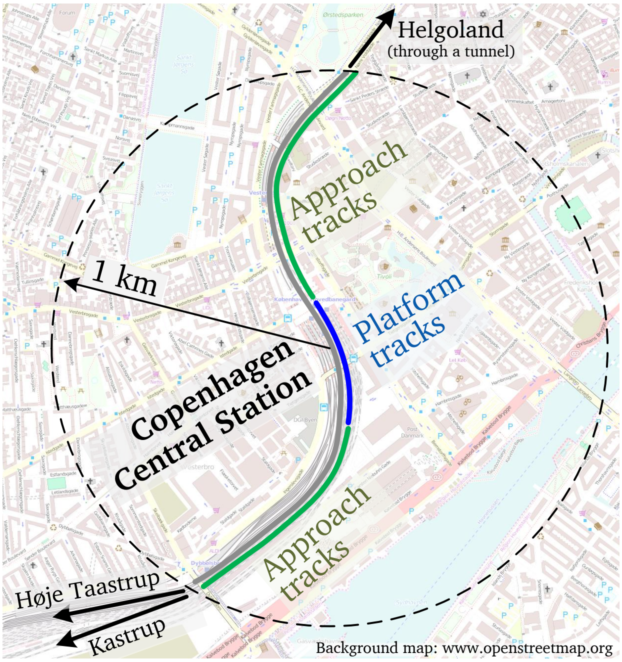


Figure 3.20: Overview of Copenhagen Central Station

Radio planning at a train station

In the Danish GSM-R network, the default cell diameter is planned to be 5 km long [19, p. 4], i.e. the average cell will have a radius of 2.5 km. However, the actual cell size must be adjusted, depending on the communication capacity demand. For instance, in a high density railway area, cells are smaller in order to serve the expected high traffic load. Due to that, at the busy Copenhagen Central Station, the radio cells should be relatively small—they should have a small radius.

Naturally, the smaller the radio cells are, the more of them must be deployed. Due to that, the train on-board radios will have to perform more inter-cell handovers. Every handover is a risk of delay and communication disruption, which may cause unnecessary slowdown of a train or even a complete stop [53, pp. 27–28]. Therefore, according to ETCS specifications [30, p. 23], handovers should be avoided in the areas where low delay of ETCS messages is required. A train station is one of such areas, because if a train is forced to stop (or even slow down) at the entry to the station, it will most likely block or delay other trains. Therefore, a communication disruption may cause a chain reaction that will lead to delay of other trains [54].

Due to that, for instance, Norwegian standards for railway radio planning [58, p. 28] recommend to avoid handovers within the stations, altogether. Also, if possible, the whole station and the 1.5 km track sections before the station should be covered by a single cell [58, p. 15]. Such radio planning is aimed at minimizing the risk of a communication disruption in the most vulnerable railway areas.

In this work, it was decided to follow this recommendation and to cover the whole station with a single radio cell. Applying this to the example of Copenhagen Central Station, the radio cell should have approximately a 1 km radius—assuming that the radio base station is placed by the platform tracks. This radio deployment will allow both the platform and the approach tracks to be covered with a single cell, as shown in Figure 3.20. In the following section, this 1 km cell is referred to as the “central cell”.

ETCS capacity requirements at Copenhagen Central Station

In order to model Copenhagen Central Station, it is required to find the number of ETCS sessions (OBU-RBC sessions) that must be simultaneously handled by the 1 km central cell. Since each train (OBU) establishes an individual ETCS session with the RBC, the number of sessions is equal to the number of trains.

Based on the analysis of the current train traffic density, the expected maximum number of ETCS sessions in the analysed area, is as follows [19, pp. 6–7, 12]:

- 12 ETCS sessions at the platform tracks, denoted as N_P.
- 34 ETCS sessions on the 5 km track section from the Central Station to Østerport (the line towards Helgoland), denoted as N_H.
- 20 ETCS sessions on the 5 km track section from the Central Station to Hvidovre (the line towards Høje Taastrup), denoted as N_T.
- 20 ETCS sessions on the 5 km track section from the Central Station to Kalvebod (the line towards Kastrup), denoted as N_K.

However, from these four track sections, only the platform tracks will be entirely covered by the central cell. In case of the 5 km track sections going in and out of the station, only part of them lies within the coverage of the central cell. Hence, only a fraction of the expected ETCS sessions has to be served by the central cell. Assuming that the trains are uniformly distributed along the tracks, the central cell will have to provide capacity for $\frac{1km}{5km} = \frac{1}{5}$ of the listed demand. This is illustrated in Figure 3.21. Therefore, the maximum number of ETCS sessions ($N_{Central}$) that the 1 km central cell must provide is found by the following expression:

$$\begin{aligned} N_{Central} &= N_P + \tfrac{1}{5} \cdot N_H + \tfrac{1}{5} \cdot N_T + \tfrac{1}{5} \cdot N_K \\ &= 12 + \tfrac{1}{5} \cdot 34 + \tfrac{1}{5} \cdot 20 + \tfrac{1}{5} \cdot 20 \\ &\approx 27\, sessions \end{aligned} \tag{3.9}$$

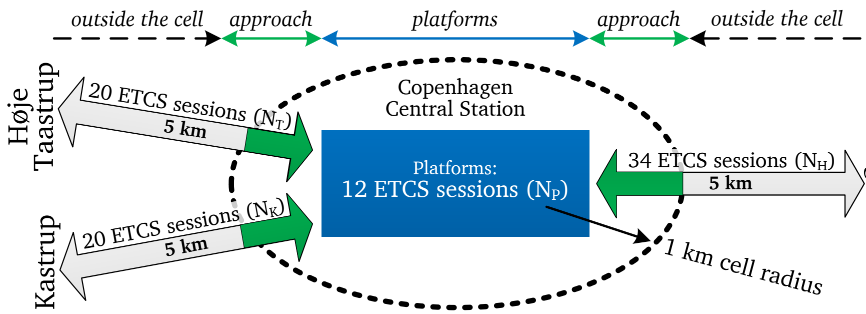


Figure 3.21: Estimation of the ETCS capacity demand at Copenhagen Central Station

Based on the above estimation, currently, the central cell at the station must provide a sufficient capacity for 27 simultaneous ETCS sessions—27 trains. However, the train traffic at the station is expected to grow in the future. In 2030, the number of trains is predicted to increase compared to 2010 by [72, p. 22]:

- 46% at the platform tracks.
- 33% on the section towards Helgoland.
- 27% on the section towards Høje Taastrup.
- 60% on the section towards Kastrup.

The demand for ETCS sessions should increase proportionally to the traffic increase. Hence, in the peak hour, the 1 km central cell will have to provide the following number of ETCS sessions:

$$\begin{aligned} N_{CentralFuture} &= 146\% \cdot N_P + 133\% \cdot \tfrac{1}{5} \cdot N_H + 127\% \cdot \tfrac{1}{5} \cdot N_T + 160\% \cdot \tfrac{1}{5} \cdot N_K \\ &\approx 39\, sessions \end{aligned} \tag{3.10}$$

A typical GSM-R cell provides capacity for approximately 23 ETCS sessions, as was explained in Section 2.4.1 on page 20. Moreover—besides ETCS—the mobile network must provide capacity for voice communication and possibly for other applications. Therefore, GSM-R may not be able to fulfil the current capacity requirements at Copenhagen Central Station. In the future, if the train traffic grows as expected, the GSM-R capacity issues will be even worse.

3.9.2 New railway applications

In railways, the communication-based applications can be classified into three categories depending on their significance for the train operation [11, p. 17]:

1. *Critical operational applications* are directly related to the train operation and its safety and/or efficiency. The best examples are: the ETCS signalling, REC and—to lower extent—other types of voice communication. Without these applications, trains either cannot operate or they operate in a degraded mode with reduced efficiency and/or safety.

2. *Business-supporting applications*—classified as non-critical—improve internal operations of a railway company. However, they are not directly related to train operation. Thus, they do not have a direct impact on the railway safety and efficiency. These applications may, for example, speed up maintenance work, improve monitoring of the rolling stock condition or provide video surveillance for security purposes. They can also deliver data to the passenger information systems, e.g. on-board screens or speakers.

3. *Infotainment applications*—also non-critical—are directly addressed to the train passengers. Examples of these applications are Internet access and movie streaming. The passengers may access them over on-board equipment, e.g. a screen built into a seat backrest. Alternatively, the applications may be accessed directly from passengers' private devices, e.g. via on-board Wi-Fi access points.

This classification is not standardized. Hence, other classifications are proposed with a slightly different division between the business-supporting and the infotainment applications [10, p. 240]. Nevertheless, it is broadly agreed in the literature that railway applications are classified into critical and non-critical [52, p. 197] [11, p. 51]. However, this classification is not black and white [33, p. 44], because the business-supporting applications indirectly affect railway operation to bigger or smaller extent. Thus, depending on operation procedures within particular railway company, criticality of some applications may be high. In this thesis, only ETCS and voice communication are considered as critical.

Currently, the GSM-R network is used almost exclusively for providing the critical applications. Less often, the network delivers also low-bitrate business-supporting applications [11, p. 17], such as passenger information displayed at station platforms. This small range of applications cannot be extended due to the GSM-R limited capacity [11, p. A-2] and its poor data transmission capabilities (see Section 2.4.2 on page 21).

Although a report by Pujol and Marcus [33, p. 13] claims that the railway companies have little interest in introducing new applications in their networks, it is rather only a consequence of the insufficient GSM-R capacity. The interest in using new applications certainly would be much higher, if the railway networks could support them. This is confirmed in an analysis by Taylor et al. [11, p. 49] and by the fact that many railway companies seek alternative solutions for delivering non-critical applications.

Besides GSM-R, railways often use a secondary communication technology or sign a roaming contract with a commercial mobile operator. For instance, UMTS, WiMAX and satellite communication are popular ways of providing infotainment applications to the passengers (e.g. the Internet access) [73]. Commercial mobile networks are often used by railway for distribution of business-supporting data among their personnel. Even though these data, such as timetable updates, working schedules or documentation, is not as critical as ETCS signalling, it is still necessary for everyday railway operation. If railway personnel do not have access to it, train operation may be severely disrupted, as illustrated by an example of Copenhagen urban railways [74]. All in all, these alternative communication solutions prove that there is a demand for new applications and that it cannot be fulfilled by GSM-R. This means that GSM-R cannot be the single communication system for all railway needs, which was one of the goals of this technology [21, p. 111].

Compared to GSM-R, LTE offers a much larger capacity and significantly improved data transmission capabilities [39, pp. 7–8]. Thanks to these, railway LTE network is expected to be able to support a range of innovative applications. Examples of such applications are remote software updates, tele-maintenance, voice announcements for the passengers, and remote cargo tracking. Since LTE is expected to be able to deliver—over a shared infrastructure—both critical and non-critical applications, all mobile communication needs of railways may be possibly satisfied with a single technology.

Proposed application mix

Assuming that LTE will become the railway communication network, it will certainly deliver some additional application besides the ETCS signalling and the voice communication. However, railway companies will choose different applications depending on their particular needs and strategy. Also, new ideas for communication-based services are expected to emerge. Hence, currently, it is impossible to predict the actual application mix that will be used in the future.

For the purpose of this analysis and the following simulations, an exemplary application mix was defined, following ideas collected from the literature. All of the proposed applications were based on communication between an on-board element (e.g. a security camera) and an application server (e.g. a security server storing video recordings), as illustrated in Figure 3.22. The proposed mix consisted of the following applications:

1. *ETCS signalling* which is based on communication between the OBU and the RBC. Details of the ETCS model were described previously in Section 3.6.1 on page 44. In the model presented here, the OBU was sending a 128-byte MA request to the RBC every 30 s, on average. The RBC was replying with MA grants that were also 128-byte long.

2. *Voice communication* (telephony) is one of the critical applications for railways [11, p. 19]. It is a necessary application for communication between drivers and dispatchers. In the presented here model, each driver was making a voice call to the dispatcher, every 600 s, on average. Each call generated two 12.2 kbit/s data streams—one in the uplink (from the cab radio to the dispatcher) and one in the downlink. A call lasted 20 s, on average.

3. *Tele-maintenance* was based on communication between the on-board sensors and the maintenance server [9, pp. 38–39]. In the uplink direction, the data collected from various sensors was uploaded to the maintenance server. It might be used to detect failure of train elements and also to plan preventive repairs. In the downlink direction, the maintenance server was sending software updates. In this model, every 900 s, the application generated two 1 MB files: one in the downlink and on in the uplink.

4. *Passenger information* provided the on-board passengers with the latest updates on delays and disruptions in traffic [10, p. 240]. The passenger information server was sending voice announcements that were played out in the passenger cabin. In this model, the application generated a 64 kbit/s downlink stream with audio message that lasted for 5 s, on average. Each of the trains was receiving an audio message every 900 s.

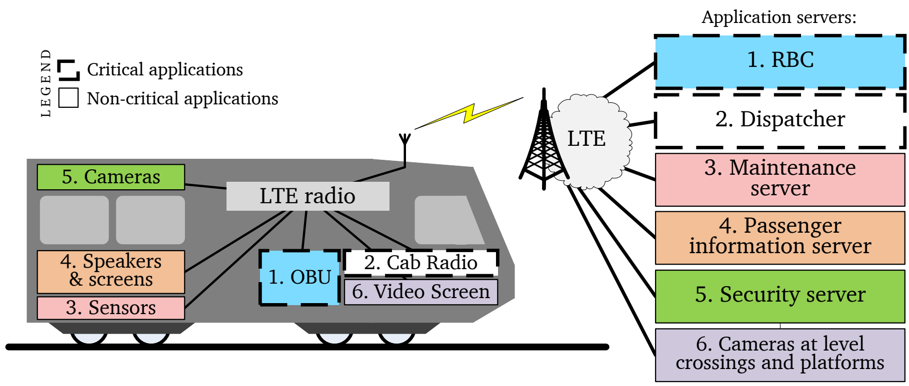


Figure 3.22: Overview of the proposed application mix

5. *Video surveillance (uplink)* provided monitoring of the train interior for security purposes [8, p. 50] [52, p. 203]. A video stream was transmitted from on-board cameras to a security server. In the model, this application generated a continuous 500 kbit/s uplink stream.

6. *Video surveillance (downlink)* provided the train driver with a live view of the potentially dangerous locations such as station platforms and level crossings [42, p. 63]. The driver was able to monitor a platform before entering a station. This allowed him/her to react if, for example, someone would fall from the platform on the track. In the model, the video surveillance application generated a continuous 500 kbit/s downlink stream.

3.9.3 Simulation model

The presented application mix was used in a simulation scenario which modelled an LTE network covering Copenhagen Central Station. The network provided connectivity between the trains at the station and a number of application servers. Similarly as in Snoghøj-Odense scenario, each train was represented as an UE. The purpose of the scenario was to investigate the following:

- Is the capacity offered by an LTE cell sufficient to support the expected number of trains at Copenhagen Central Station?
- How the growing number of trains at the station (the growing traffic load) affect the ETCS transmission performance in terms of ETCS transfer delay and ETCS data loss?
- How is the ETCS transmission affected by the non-critical applications? Is it possible to provide both the critical and the non-critical applications over a shared network.

In the scenario, a single LTE cell—a single eNodeB—provided radio coverage over the whole station, as shown in Figure 3.23. All of the UEs were connected to that cell, which had approximately a 1 km radius. Additionally, four jammer nodes were introduced (two uplink and two downlink jammers). Their purpose was to model interference from the neighbouring LTE cells. Besides these, the network model consisted of the EPC node and the application servers.

The modelled network operated in a 5 MHz bandwidth at the 921 MHz carrier frequency. The eNodeB antenna height was 40 m. In the whole cell, the received signal power was above −92 dBm, fulfilling the respective ETCS requirement. The remaining parameters were configured the same as in the Snoghøj-Odense scenario (see Table 3.3 on page 52).

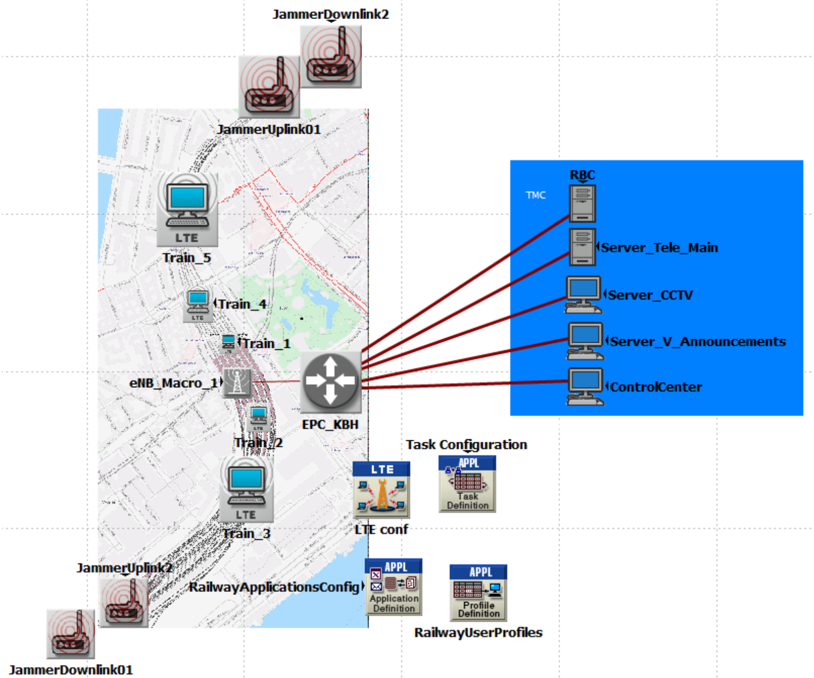


Figure 3.23: Model of the LTE network deployed at Copenhagen Central Station. The case with 5 trains is shown.

It must be noted that—in reality—the tracks to the North of the station are hidden inside of a tunnel. However, in the simulations, this was neglected and it was assumed that the radio signal is provided there by a proper mechanism.

The simulation scenario was considered in 11 cases that differed in the number of trains (UEs) present at the station. The investigated range was from 5 to 70. The radio cell at the station must currently provide capacity for 27 trains and for 39 trains in the future (see Eq. 3.9 and 3.10 on page 71). However, in order to investigate the ETCS transmission behaviour at higher traffic load, up to 70 trains were considered in the simulation scenario.

Train distribution

In the scenario, all trains (UEs) were modelled as stationary nodes. Such a setup was chosen, because—in a station area—most of the trains are stopped by the platforms or run at a relatively low speed. Therefore, speed is a minor factor affecting ETCS transmission. The trains were distributed uniformly along the station tracks.

Quality of Service (QoS) provisioning in LTE

The proposed application mix, which was described in Section 3.9.2, consists of both critical (ETCS and voice communication) and non-critical business-supporting applications (the remaining four). Both types share the same network resources. The critical applications are vital for railway operation, so their transmission performance cannot be affected by the non-critical applications. Due to that, a QoS provisioning mechanism is required. Its purpose is to prioritize and protect the critical applications.

In LTE, the QoS provisioning is provided using EPS bearers, which carry packet flows between the UE and a specific P-GW [38, p. 25]. Every UE has at least one EPS bearer, which is called the *default bearer*. It is automatically established during the UE network attachment procedure and it provides best-effort service.

If an application requires specific transmission performance, then a *dedicated bearer* can be established for that application. Every dedicated bearer has an associated set of QoS requirements. By transmitting a packet flow over a dedicated bearer, it is possible to guarantee certain QoS performance for that flow [38, p. 34]. For instance, the packets carrying ETCS messages can be transmitted over a dedicated bearer with a guaranteed minimum bitrate and low delay budget. Since EPS bearers are established between the UE and the P-GW, the transmission performance is guaranteed across the whole LTE network. Every EPS bearer is characterized by three parameters [75, Sec. 6.1.7]:

- *QoS Class Identifier (QCI)*. There are nine QoS classes defined in LTE Release 8 (additional classes have been defined in Release 12). Each class has a predefined set of transmission requirements regarding: scheduling priority, delay budget, and packet loss rate.
- *Guaranteed Bit Rate (GBR)* defines the minimum bit rate that must be guaranteed for the bearer by the network regardless of the traffic load. Alternatively, a bearer can be non-GBR—for which the network do not guarantee any resources. Therefore, a non-GBR bearer can be affected by congestion. Type of the bearer (GBR or non-GBR) is predefined for each QCI.
- *Allocation and Retention Priority (ARP)* determines if a newly requested bearer can pre-empt an already established bearer. It is used when the network resources are not sufficient to establish all bearers demanded by UEs. ARP does not affect packet scheduling priority, which depends on the QCI.

EPS bearers were used in the simulation scenario for providing the necessary protection of ETCS traffic. Table 3.4 presents the chosen bearer configuration. Besides the default bearer, two dedicated bearers were defined for: the ETCS and the voice communication. The bearer delivering ETCS messages was assigned QCI 3, in order to ensure high scheduling priority and a low delay budget. QCI 3 is a GBR bearer. This configuration followed a recommendation by Khayat et al. [52] that a train control system should be provided over a GBR bearer.

Table 3.4: EPS bearer configuration for ETCS simulations

Bearer:	**ETCS**	**Voice**	**Default**
QCI	3	2	9
Scheduling priority†	3	4	9
Delay budget†	50 ms	150 ms	300 ms
Packet loss rate†	10^{-3}	10^{-3}	10^{-6}
GBR Uplink	32 kbit/s	64 kbit/s	—
GBR Downlink	32 kbit/s	64 kbit/s	—
ARP	1	2	3

† Pre-configured for a given QCI, as defined in [75, Tab. 6.1.7]

3.9.4 Simulation results

This section presents the results collected is the simulation scenario modelling Copenhagen Central Station. Each of the 11 simulation cases was executed 15 times with different seed numbers. A single simulation run lasted for 15 minutes. Additional details on the simulation scenario are presented in Appendix B on page 183.

Radio throughput and utilization

Firstly, the radio throughout results are analysed. Figure 3.24 shows the mean radio throughput in relation to the number of trains at the station. Only the uplink throughput is shown, because the downlink throughput was virtually the same.

In the case with 5 trains, the mean radio throughput was 2.2 Mbit/s. As more trains (UEs) were placed at the station, the throughput increased proportionally. This continued until the case with 35 trains, when the throughput reached a saturation at 13.2 Mbit/s. It did not increase further, despite the growing number of trains. As visible in Figure 3.25, the utilization of the radio physical channels approached 100% in the case with 35 trains*. Therefore, the traffic load exceeded the available radio capacity.

With the proposed application mix, the modelled LTE cell offered sufficient capacity to carry the traffic load generated by 30 trains. The main sources of the

*The utilization results shown in Figure 3.25 are lower than the actual radio utilization. This is because they were computed over the whole duration of a simulation run—including network initialization and protocol convergence, during which only minimal traffic was transmitted. However, during ETCS operation, the actual utilization was 100%.

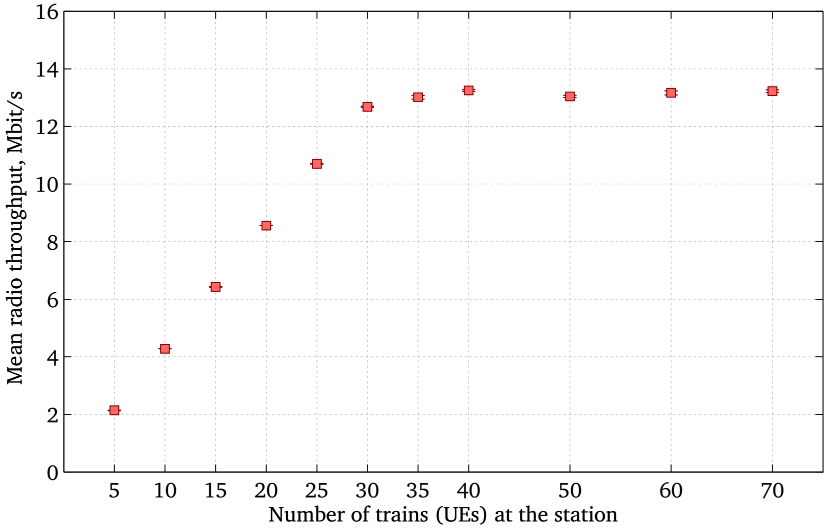


Figure 3.24: Mean uplink radio throughput in relation to the number of trains at the station. The mean downlink throughput was virtually the same. Error bars indicate 95% confidence intervals.

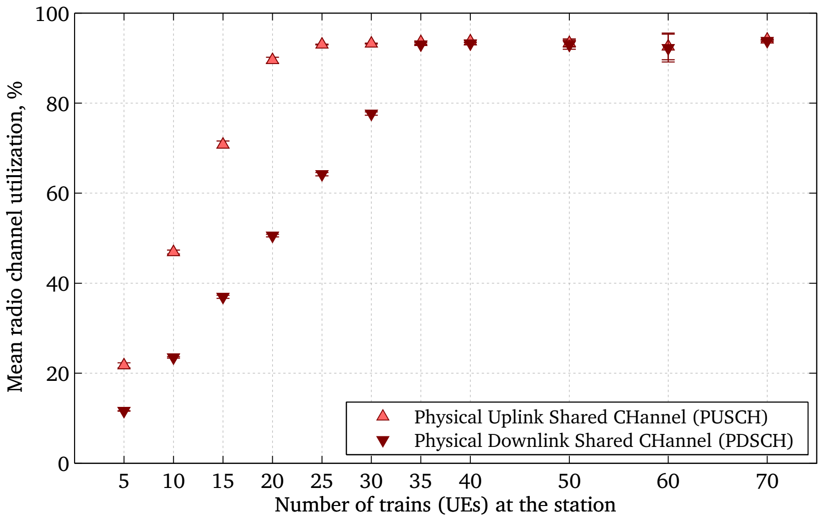


Figure 3.25: Mean radio channel utilization in relation to the number of trains at the station. Error bars indicate 95% confidence intervals.

traffic were the two video surveillance applications. Therefore, if the application mix was modified, for instance, by abandonment of the video applications, the cell capacity would be sufficient to support more trains.

ETCS transfer delay

Despite the traffic load that exceeded the radio capacity, ETCS messages were still delivered between the OBUs and the RBC. Therefore, the communication necessary for ETCS operation was provided in all of the considered cases. Nevertheless, it had to be investigated how this traffic load affected the OBUs-RBC communication. Thus, Figure 3.26 presents the mean ETCS transfer delay in relation to the number of trains at the station. The results show the end-to-end delay—measured between the OBU and the RBC.

Firstly, the *downlink* transmission was considered, i.e. from the RBC to the OBU. In this direction, the mean transfer delay was approximately 2 ms in all of the investigated cases. Thus, the number of trains at the station and the consequent traffic did not affect the downlink delay.

Secondly, the *uplink* transmission was considered. In the first cases—as long as no more than 30 trains were present in the cell—the mean uplink delay was 20 ms. However, once the number of trains exceeded 30, the uplink delay began to increase proportionally to the number of trains. It reached a maximum of 32 ms in

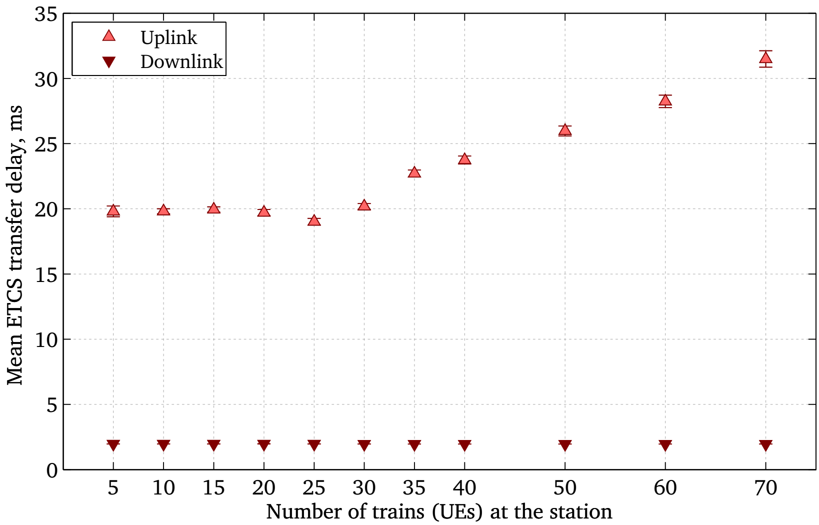


Figure 3.26: Mean ETCS transfer delay in relation to the number of trains at the station. Error bars indicate 95% confidence intervals.

the case with 70 trains. The difference between the uplink and downlink delay is explained in the following section.

From the ETCS perspective, these difference in transfer delay between the uplink and downlink directions was insignificant. Similarly, the delay increase due to the number of trains was also insignificant. This is because all of the observed results were approximately 25 times smaller that the maximum acceptable 500 ms (see Table 3.2 on page 42). Moreover, the maximum recorded delay was 546 ms. Thus, 100% of ETCS messages were delivered within 1.5 s. Therefore, the modelled LTE network provided ETCS transmission with performance that is significantly better—in terms of delay—than required by railways.

Difference between the uplink and the downlink

Looking again at Figure 3.26, it is visible that the uplink delay was approximately 10 times higher than the downlink delay. Due to that, the uplink delay would reach the maximum acceptable ETCS delay limit before the downlink delay would. Thus, the uplink delay should be considered as the limiting one.

The ETCS transfer was slower in uplink due to the specifics of the uplink radio resource scheduling. In LTE, both the uplink and the downlink packet schedulers are placed in the eNodeB. This means that the downlink scheduler is collocated with the downlink packet buffers (queues). Since both elements are in the eNodeB, the scheduler has instant information about the downlink buffer status.

Contrary to the downlink case, the uplink scheduler and the uplink packet buffers are separated. The uplink scheduler is located in the eNodeB, while the uplink buffers are distributed among the UEs [76]. Due to that, there is an inevitable delay between the moment a new uplink data is placed in the buffer (in one of the UEs) and the time the uplink scheduler (in the eNodeB) is informed about this data. The uplink scheduling procedure is as follows [38, pp. 114–115, 372]:

1. When the UE has new data to send in uplink, it needs to notify the eNodeB. This can be done in three ways. Firstly, if PUSCH resources are available, the UE uses them to send a Buffer Status Report (BSR). Alternatively, the UE may send a Scheduling Request (SR) over the Physical Uplink Control CHannel (PUCCH). Finally, if the UE has resource on neither of these channels, the SR may be sent over the PRACH. However, this includes a risk of a collision with other UEs.

2. When the a BSR or an SR arrives to the eNodeB, the uplink packet scheduler becomes aware of the data waiting at the UE.

3. The scheduler assigns the uplink radio resources and distributes Scheduling Grants (SGs) via the Physical Downlink Control CHannel (PDCCH).

Since this signalling exchange between the UE and the eNodeB is a necessary step in uplink scheduling, the uplink delay is always greater than the downlink delay. Especially when the SR must be transmitted over the random access procedure, the scheduling delay becomes significant.

Furthermore, the SR and the BSR do not carry the full information about the uplink data waiting at the UE. The SR informs that there is some data waiting, but it does not specify its size or the EPS bearer that it belongs to. The BSR contains more information: an approximate size of the data and the Radio Bearer Group (RBG) that the data belongs to. A RBG gathers EPS bearers with similar QoS requirements [39, pp. 197–198]. Since, the BSR reports buffer status per RBG—not per single bearer—the information available to the eNodeB scheduler is only approximate. In the herein used simulation model, all GBR bearers belonged to a single BSR. Thus the scheduler was not aware if the waiting uplink data belongs to the ETCS bearer or to the voice bearer—both of which were GBR. All in all, the uplink scheduling is more complex than the downlink. The information available to the scheduler at the eNodeB is always slightly delayed and approximated.

The uplink scheduling procedure is also the reason why the ETCS uplink delay began to increases in the cases with more than 30 trains. As it was explained, the uplink scheduling requires resources on the control channels, namely the PUCCH, the PRACH, and the PDCCH. Therefore, capacity of the control channels is one of the main factors affecting the uplink delay [77]. In the overload cases—with more than 30 trains—a congestion on the control channels caused a delay in the SR and the BSR delivery. Consequently, the uplink ETCS messages were delayed as well. Therefore, despite the guarantees provided by the EPS bearer, the uplink ETCS traffic was affected by the traffic load.

ETCS data loss

ETCS data loss was not observed in any of the investigated cases. Similarly as in the previous scenarios, this was due to the retransmission mechanisms on the radio and on the application layers. On average, 0.07% of the ETCS messages had to be retransmitted. The retransmission rate was not affected by the number of trains (UEs) at the station.

ETCS data integrity in LTE and mechanism for minimizing data loss probability are investigated in detail in Section 5.4 on page 132.

QoS mechanism performance

The EPS bearer dedicated to ETCS had a delay budget of 50 ms (see Table 3.4 on page 78). Both the uplink and the downlink mean delays remained within this budget—in all of the investigated cases. Even when the offered traffic exceeded the available radio capacity, the ETCS transfer delay remained low. Therefore, the bearer-based QoS mechanism fulfilled its goal of prioritizing ETCS traffic.

This is very important from the railway point of view. It means that the LTE network is able to provide simultaneously both critical and non-critical applications over shared network resources. Taking also into account the high transmission capacity of LTE, an opportunity opens for the railways to introduce new communication-based applications. These could improve railway operation (e.g. tele-maintenance), increase safety (e.g. video surveillance), or offer new services to the passengers. Regardless of the traffic generated by these new applications, the LTE network provides timely and reliable ETCS communication.

Furthermore, the effective QoS mechanism opens a possibility of delivering railway applications over commercial (public) LTE networks. This possibility—in GSM-R scenarios—was investigated by Del Signore et al. [78]. In a commercial LTE network, railway traffic could be separated and protected from other traffic by the dedicated EPS bearers. However, roaming in commercial networks requires further investigations, which were out-of-scope of this thesis.

Capacity increase compared to GSM-R

The modelled LTE cell provided enough resources to fulfil the current (27 trains) and the future (39 trains) ETCS demand at Copenhagen Central Station. Moreover, the cell could provide OBU-RBC communication for at least 70 trains. Therefore, LTE offers sufficient capacity even for an unlikely scenario that the train traffic in Copenhagen increases twice as much as expected.

This high capacity makes LTE a relatively safe investment for railways, because it guarantees that—even in the limited 5 MHz bandwidth—the network will not become an obstacle limiting ETCS operation as GSM-R became. Compared to the current GSM-R network, which can support approximately 23 trains per cell (see Section 2.4.1 on page 20), an LTE network offers at least a threefold capacity increase in terms of trains per cell.

3.10 Chapter conclusions

In the field of railway signalling, ETCS is one of the most important developments of the recent years. The system improves safety and efficiency of train operation. Moreover, it is a revolutionary standard in terms of international interoperability. Due to that, ETCS became an essential element of modern railways.

ETCS is a communication-based system. Therefore, its performance depends on the supporting mobile communication technology. Currently, ETCS is provided over the outdated GSM-R. Due to the insufficient capacity and inefficiency of GSM-R, alternative technologies are considered to replace it in the future.

The work that was presented in this chapter investigated a hypothesis that LTE can become the future railway communication technology. This means that LTE must fulfil ETCS transmission requirements. Also, LTE must address the problem of insufficient capacity of the current railway network. In order to validate this hypothesis, ETCS transmission performance over LTE was investigated in different simulation scenarios, which modelled typical railway conditions.

The simulations results showed that LTE is able to offer ETCS communication—between the OBU and the RBC—that fulfils the requirements in terms of transfer delay and data loss. The simulations investigated also the impact of base station deployment density, train speed, and traffic load on the ETCS transmission performance. Although these factors affected ETCS communication, neither of them prevented the LTE network from fulfilling the requirements.

Certain elements of LTE, such as the random access procedure and the uplink scheduling request procedure, are suboptimal for the low-rate ETCS traffic. Despite that, LTE offers delay performance that is significantly better than required by ETCS system. The low delay, which is considerably lower than in GSM-R, could be exploited by railways in the future versions of ETCS. For instance, thanks to the low delay, it might be possible to reduce level crossing closing time or decrease the headway time on the moving-block-based railways (ETCS Level 3). However, the impact of ETCS delay on train operation should be investigated in future work, since it was out of scope of this research.

The simulation results also demonstrated that the capacity of an LTE cell—in terms of the number of supported ETCS-equipped trains—is significantly higher that the capacity of a GSM-R cell. The reason is that LTE offers much higher throughput and efficiency, thanks to packet-switched transmission, advanced modulation, and flexible resource allocation. LTE can solve the problem of insufficient capacity in GSM-R networks. LTE is able to fulfil both the current and the future capacity demand at Copenhagen Central Station, which is the most challenging railway area in Denmark due to the train traffic intensity. Therefore, LTE-based railway communication network would not be a bottleneck limiting railway operation as the GSM-R network may be.

Finally, thanks to the bearer-based QoS mechanism, the LTE capacity can be used for providing both critical and non-critical applications. Therefore, even in overload conditions, the modelled LTE network provided good ETCS transmission performance. The railway mobile network based on LTE can provide many more applications besides ETCS. Hence, LTE—by delivering new innovative applications—creates an opportunity to enhance railway operation, improve safety, increase security, and enrich passenger experience.

CHAPTER 4

Railway voice communication in LTE

In railway, the data-based applications are continuously growing in terms of importance and popularity [11, p. 5]. The best examples are the modern command-control systems, such as ETCS and CBTC. They became fundamental elements that are necessary for safe, efficient, and high-speed railways.

Despite that, voice communication is still a crucial feature from the railway point of view [11, p. 19]. Voice calls are used in everyday operational procedures, such as ETCS mission start-up [53, p. 43] and shunting [22, p. 33]. Besides, voice communication shows its special importance in case of extraordinary and unplanned situations. For example, if a train stops in an unexpected location, ETCS can only inform the dispatcher that the train is not moving. Without a voice call from the driver, the dispatcher cannot know why the train stopped: due to a technical failure, due to an obstacle on the track, or due to an accident. Hence, voice communication does not become redundant with the introduction of ETCS. On the contrary, voice communication is a feature that complements even the most advanced railway signalling system.

Voice communication is used by almost the entire railway personnel. Besides drivers and dispatchers, it is also necessary for track-side, maintenance and other employees, who are often distributed over very large areas. For them, voice communication is an everyday work tool. Therefore, even if the role of voice communication may change over time [11, p. 49], it is unlikely to diminish significantly.

Because of the above reasons, the future railway mobile network must satisfy specific railway needs related to voice communication. These include both feature and performance requirements. Therefore, regardless of the data-transmission performance, LTE must fulfil also railway voice requirements in order to be considered a viable alternative to GSM-R. This is a challenge, because LTE has been designed as a network for data-based applications. This is in contrast to GSM which has been designed and optimized for delivering telephony. The telecommunication industry has recognized the need for a VoLTE standard relatively late [79], i.e. only after the original LTE standard has been established. As a result, VoLTE [80], which is a Voice over IP (VoIP) standard based on IP Multimedia Subsystem (IMS) [81], gained a broad support of the telecommunication industry only recently.

The availability of voice communication in LTE was discussed by Liem and Mendiratta [41]. Then, Calle-Sanchez et al. [40] proposed using VoLTE for providing railway voice communication in a railway LTE network. In a later publication by the same authors [10] and in a publication by Zayas et al. [42], it has been claimed that railway voice features can be built using standard VoLTE and LTE mechanisms. Nevertheless, these research works presented only theoretical considerations. Hence, the actual performance of VoLTE in the railway environment must be validated and confronted with railway requirements. This challenge constitutes the main motivation for the research work presented in this chapter. The specific goals of the work were defined as follows:

- Discuss railway voice communication requirements.
- Propose how the critical railway voice communication features, namely the operational one-to-one call and the Railway Emergency Call (REC), can be built in VoLTE.
- Validate VoLTE performance—in terms of call setup time, voice packet delay, and voice packet loss—using a simulation-based approach.
- Compare VoLTE performance with the voice requirements defined by the railway industry.

Chapter organization

This chapter is organized as follows. The next section presents the specifics of railway voice communication in comparison to the commercial mobile telephony. Then, VoLTE architecture and call setup procedures are presented. At last, VoLTE performance is validated in simulations using extended scenarios from Chapter 3.

The research work presented in this chapter has been previously published in a paper [Sniady2015a].

4.1 Railway voice communication requirements

Voice communication is an essential tool, which is necessary in the work of almost all railway personnel: train drivers, dispatchers, shunting staff, maintenance staff, on-board personnel, and others. They all communicate using a variety of terminals, which can be classified into three main types [12, pp. 154–157]:

- *Cab-radios*, which are the voice terminals built into the train driver desks.
- *Fixed terminals*, which are used, e.g. by the dispatchers. These terminals are connected over a fixed network.
- *Handheld radios*, which are used for operational, shunting, and other purposes [23, p. 31]. These terminals are usually similar to mobile phones known from the commercial telephony.

Considering its core functionality, the voice communication for railways is similar to the voice communication offered by the commercial mobile telephony. For example, according to the UIC requirements [23, pp. 32–33], railway radios must support many features known from the commercial networks, such as:

- One-to-one calls,
- Caller identity display,
- Call forwarding,
- Call hold.

Due to these similarities in functional requirements, railways often can reuse the communication standards known from commercial telephony. The best example is GSM-R, which is almost entirely based on the commercially used GSM [12, p. 148]. Despite these similarities, railway voice communication has its own specifics. Thus, the communication technology for railways may have to be enhanced with additional features and optimizations. Besides, railways impose specific performance requirements in terms of call setup time. These differences to the commercial mobile telephony are described in the following section.

4.1.1 Railway-specific voice features

Group and broadcast calls

One-to-one voice call (also called *point-to-point* call) is the basic call type that must be supported by the railway network. However, in their everyday operations, railways also use other call types, which are usually not offered in the commercial telephony. Therefore, the communication network must additionally support the following call types [23, pp. 21–22]:

- *Broadcast calls*, including the REC, which are used to reach all terminals within a particular area. Usually, only the call initiator is allowed to speak.
- *Group calls*, which are used for communication within a predefined group of users, e.g. train drivers.
- *Multi-party calls*, which are similar to group calls, but the call parties are chosen ad hoc during call initiation.

REC is an especially important call type [11, p. 19], because of its impact on railway safety. REC is a broadcast call that is used only in case of a dangerous situation. It can be initiated with a press of a dedicated button on any railway voice terminal. The call initiator is then automatically connected to the dispatcher responsible for the particular railway area. The conversation between the initiator and the dispatcher is broadcasted to all other terminals within the area. In this way, everyone is immediately informed about the danger.

In order to ensure that REC is received by all terminals, this call type is given the highest priority by the network. REC pre-empts any other voice communication and, also, it can pre-empt any data communication, including ETCS traffic. Moreover, REC is automatically answered by the terminals without the need for any reaction from the user. Such a solution guarantees that all users listen to the ongoing REC.

Call addressing

In order to simplify and speed up voice communication, railways introduced two features related to the call addressing:

- *Functional Addressing (FA)* provides automatic translation between a railway function (e.g. "dispatcher") and the corresponding phone number. This allows the caller to use the function instead of the phone number while placing a call. For instance, in order to call a driver of a particular train, it is sufficient to enter the train running number. Thanks to this, the train driver can be easily reached regardless of who is the driver on that day or which train unit is used.

- *Location Dependent Addressing (LDA)* redirects the call depending on the current caller location. LDA is used most often when a train driver calls to a dispatcher. Then, LDA automatically selects the particular dispatcher responsible for the area where the train is currently located.

Call prioritization

Railway communication network carries different call types. From the point of view of safety and train operations, they have different importance. Therefore, the network must provide a mechanism that assigns appropriate priority depending on the call type. The high-priority call should pre-empt the lower priority calls [23, p. 21]. For example, REC should pre-empt a one-to-one call.

4.1.2 Performance requirements

Apart from the additional features, railways impose their own requirements on the call setup time [23, pp. 27–28]. The maximum acceptable setup time is defined for each call type separately, as summarized in Table 4.1. The strictest requirement applies to REC, because it is a critical call used in the extraordinary and dangerous situations. A fast REC setup may prevent accidents.

Table 4.1: Railway call setup time requirements [23, p. 28]

Call type	Setup time
Railway Emergency Call (REC)	< 2 s
Group calls between drivers in the same area	< 5 s
Other operational mobile-to-fixed calls	< 5 s
Other operational fixed-to-mobile calls	< 7 s
Other operational mobile-to-mobile calls	< 10 s
All low priority calls	< 10 s

The required call setup times shall be achieved in 95% of cases. Call set-up times for 99% of cases shall not be more than 1.5 times the required call setup time.

Once a call is established, the network begins to transmit voice frames between the call parties. The performance of this transmission affects the quality of the received voice and, therefore, the usability of the voice communication. The voice transmission is affected by the transfer delay and the frame loss in the underlying network [82, p. 5]. However, railways do not impose any specific requirements on these measures or the resulting received voice quality. Therefore, in the work presented in this chapter, it was assumed that the railway mobile communication network must fulfil the same voice transmission requirements as any other network delivering VoIP services.

The total end-to-end delay, i.e. so-called "mouth-to-ear" delay, consists of the transfer delay and the coding/decoding delay (including all processing in the end user terminals). If the "mouth-to-ear" delay is below 150 ms, then the listeners do not notice distortions in the received voice. Moreover, even delays up to 200 ms do not cause annoying effects [83]. For the purpose of the following analysis, it was assumed that the coding/decoding delay can be up to 50 ms, which is in accordance with an example published by Holma and Toskala [39, p. 262]. Hence, in order to have the total "mouth-to-ear" delay below 200 ms, the maximum acceptable transfer delay is 150 ms.

The acceptable frame loss depends on the chosen codec. In this work, it was assumed that the network uses Adaptive Multi-Rate (AMR) codec fixed at 12.2 kbit/s. In order for AMR to provide good voice quality, the maximum acceptable frame loss is approximately 1% [82, p. 5].

4.2 Voice over LTE (VoLTE)

LTE is the first fully packet-switched mobile communication network [25, p. 206]. Thanks to this, the network architecture is simpler. On the other hand, it means that LTE does not include the circuit-switched network part, which was traditionally used for voice communication in the previous generations of the 3GPP mobile standards, e.g. GSM and UMTS [25, p. 206].

Circuit-Switched Fall Back (CSFB)

In order to overcome the lack of a standardized voice communication in LTE, 3GPP proposed the Circuit-Switched Fall Back (CSFB) as a temporary solution. CSFB is a procedure that forces an LTE terminal (UE) to switch to a GSM or an UMTS network in the event of an incoming or an outgoing voice call [84, p. 252]. This means that, in order to receive or make a voice call, an UE must turn off its LTE radio and handover all ongoing communication to one of the legacy networks.

CSFB has multiple disadvantages, such as a relatively long call setup time and the discontinuity of data communication during a voice call [84, p. 253]. These are especially problematic for railways for two reasons. Firstly, the fast call setup is one of the railway priorities. Secondly, the data-based ETCS signalling would have to be handed over between the networks every time a voice call is established or torn down. These inter-network handovers would increase the ETCS transfer delay. Besides, CSFB requires maintaining one of the legacy networks only for voice communication provisioning. Therefore, although CSFB is broadly used in the commercial LTE networks, it cannot be considered as a desired solution for the railway LTE network.

VoLTE as the telecommunication industry standard

Since the CSFB shortcomings are also problematic for commercial operators and their customers, there was a need to develop a better voice communication solution for LTE [85, p. 2]. Multiple alternatives emerged, such as the IMS-based VoLTE (initially called *One Voice* [79]), Voice over LTE via Generic Access (VoLGA), Simultaneous Voice and LTE (SV-LTE), and Over The Top (OTT) solutions [84, p. 251]. Among these competing standards, the IMS-based VoLTE have few important advantages, such as:

- It requires neither the legacy radio networks, nor the legacy circuit-switched core network.
- It does not interrupt data communication over LTE.
- It is based on well-defined open standards and provides an inter-operable solution.
- It offers call supplementary features, such as call waiting, forwarding, etc.

Due to its advantages, VoLTE gained support of the Global System for Mobile Association (GSMA) and was also backed by a significant number of mobile operators and equipment vendors. Therefore, VoLTE became the industry standard for voice communication in LTE [85, p. 3]. As a consequence, VoLTE should be also considered as a possible candidate for providing railway voice communication in LTE.

VoLTE is based on the IP Multimedia Subsystem (IMS) [81], which is a standardized IP-based architecture for the access-independent delivery of multimedia services. IMS is based on a set of well-defined open protocols [85, p. 23]. Another strength of IMS is that it provides a broad range of standardized functionalities for management of the IP-based services, such as: user roaming, inter-working with circuit-switched networks, and QoS negotiation. These advantages are inherited by the VoLTE standard, which defines a subset of the IMS functionalities that are necessary for providing an inter-operable telephony service.

4.2.1 Architecture

VoLTE architecture, as shown in Figure 4.1, consist of three main parts: E-UTRAN, EPC and IMS. The first two parts are standard elements of the LTE architecture, as described in Section 3.3 on page 37. The third part, namely the IMS, is responsible for call setup and call management.

The central elements of IMS are the Call Session Control Functions (CSCFs), which provide user registration, session (call) establishment, signalling routing and session management. CSCFs functionality can be split into four separate logical elements [85, pp. 32–35]:

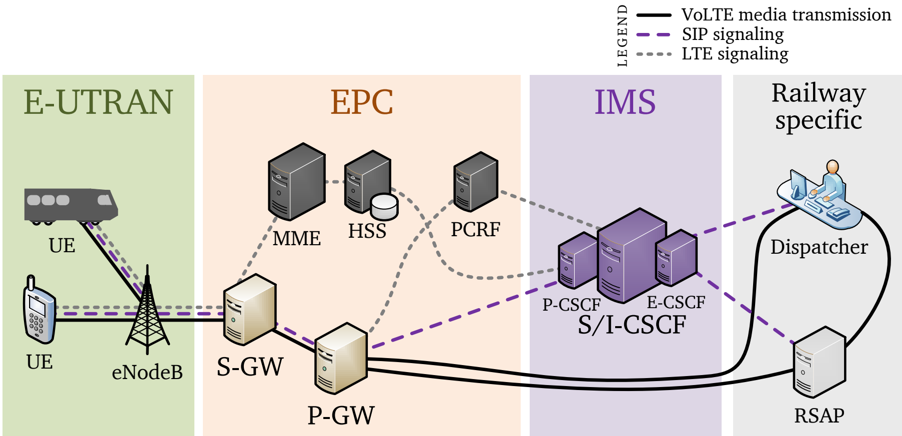


Figure 4.1: Simplified VoLTE architecture, which may be divided into the LTE radio part (E-UTRAN), the LTE backbone (EPC), and the IMS backbone. In the railway environment, two additional elements are added: the Dispatcher and the Railway Safety Answering Point (RSAP).

- *Proxy CSCF (P-CSCF)*, which is responsible for authorizing resources, detection of emergency sessions, signalling compression, and communication security.
- *Serving CSCF (S-CSCF)*, which is responsible for user registration, authorization, and call routing.
- *Interrogating CSCF (I-CSCF)*, which is used as the contact point for sessions incoming from external IMS domains.
- *Emergency CSCF (E-CSCF)*, which is responsible for routing emergency calls to the correct Public Safety Answering Point (PSAP).

Besides CSCFs, the full IMS architecture consists of other logical elements, many of which are responsible for supporting functions, e.g. inter-working with external networks and call charging. However, in this work, only those elements that are directly involved in the call setup procedures are of interest.

Finally, the last important element of the VoLTE architecture is the *Public Safety Answering Point (PSAP)*, which handles emergency calls. In this work, the considered network is dedicated for railways. Therefore, it is proposed to replace PSAP with the railway-optimized *Railway Safety Answering Point (RSAP)*. This new node, besides the standard PSAP role, provides also functionality necessary for handling RECs.

In the two previously mentioned publications [10, 42], it is explained how VoLTE can provide railway-specific voice features using a combination of various mechanisms and protocols, such as: LTE Localization Services, Push-to-talk over Cellular (PoC), Session Initiation Protocol (SIP) addressing, and EPS bearer-based QoS mechanism. Therefore, in this work, it was decided to focus on VoLTE performance in terms of call setup time and voice transmission. Two railway features, namely one-to-one call and REC, were chosen to be investigated due to the reasons explained in the following sections.

4.2.2 VoLTE one-to-one call setup

The operational one-to-one call is the first call type that was chosen, in this research work, to be modelled as a VoLTE session. This is the call type that is the most often used in everyday railway operation.

From the railway point of view, the most important element are the call setup procedure and the time it takes to complete it. In VoLTE, calls are established using Session Initiation Protocol (SIP) and Session Description Protocol (SDP) [86]. SIP provides the means for session initiation, control, and termination. SDP is used for defining media transmission and its parameters, e.g. codecs and IP addresses.

VoLTE call setup procedure is illustrated with a SIP message flow shown in Figure 4.2. There are three entities involved in the SIP message exchange: the call initiator, the called party, and the CSCF, which is routing the signalling messages between the two end points. In the presented example, only one integrated CSCF is assumed. This is because, due to a relatively small size of the railway network, the functionality of the four logical CSCFs can be placed in a single node.

Assuming that the call initiator (in this example an UE) is already registered in the LTE and the IMS networks, it can initiate the call with a *SIP INVITE* message (all the messages mentioned in this section are SIP). The message is sent from the initiator to the CSCF. It contains all information necessary for the call setup, such as: the called party identification, IMS network identification, and the call identification. Moreover, the *INVITE* includes also a description of the desired media flow in terms of codecs, IP addresses, and port numbers. This media description is written in SDP format [85, pp. 120–129].

The *INVITE* is sent to the CSCF, which replies to the call initiator with a *TRYING* message. It informs that the *INVITE* was received and is being handled. Then, CSCF resolves the IP address of the called party, i.e. it translates the called party identification (e.g. a phone number) to an IP address. This address resolution may involve contacting a Domain Name System (DNS) server. However, in the analysed case, both end points are in the same IMS domain, so DNS is not involved. After resolving the address, the *INVITE* messages is forwarded to the called party, i.e. the call destination.

After receiving the *INVITE*, the called party replies with a *SESSION PROGRESS* message in order to notify the caller that the call invitation was received. Moreover,

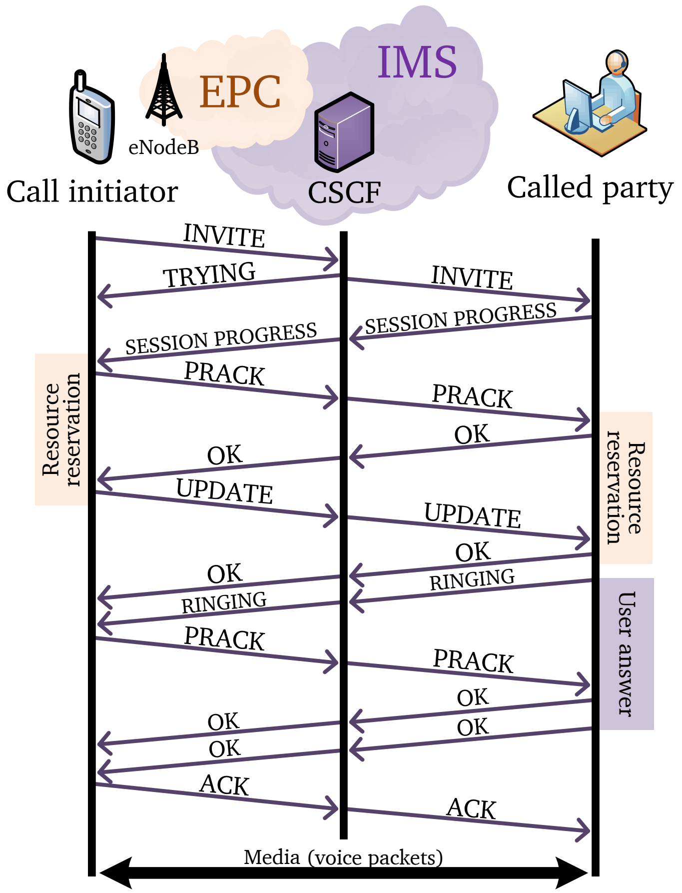


Figure 4.2: SIP message exchange during the VoLTE one-to-one call setup. An example of a mobile-to-fixed call is presented.

the *SESSION PROGRESS* message also carries an SDP reply with the information on which codecs are accepted by the called party [85, pp. 127–129].

When this initial message exchange is completed, the CSCF sends a resource reservation request to the PCRF. Resources, in the form of EPS bearers are established by the EPC and E-UTRAN nodes, i.e. P-GW, S-GW and eNodeB. The call end points exchange *PRACK* and *OK* messages in order to inform about the ongoing resource reservation process. When the EPS bearers are established, *UPDATE* and *OK* messages are exchanged [87, pp. 109–113].

Once the resources are reserved, the called party terminal notifies the user about the incoming call. At the same time, a *RINGING* message inform the call initiator that the called party is waiting for the user answer. When the user answers the incoming call, the end points exchange *OK* and *ACK* messages. Then, the media flow begins.

4.2.3 VoLTE REC setup

REC is the second call type that was chosen to be modelled in this research work. Among the various railway-specific voice features, REC is the most requested by railways [11, p. 19]. Moreover, from the communication point of view, it is also the most challenging, because REC requires fast setup, which, at the same time, involves many network nodes and user terminals.

REC is a unique railway feature, which does not have an equivalent in commercial mobile networks. Hence, it is an open question how to implement REC in VoLTE. In this work, REC is proposed as a set of multiple interrelated one-to-one calls. Although, CSCF is still responsible for routing signalling for each of the individual one-to-one calls, it is the RSAP that combines them into a REC. All of the individual calls are established between the RSAP and: the REC initiator, the dispatcher, and the listening terminals. Figure 4.3 shows the proposed REC setup procedure.

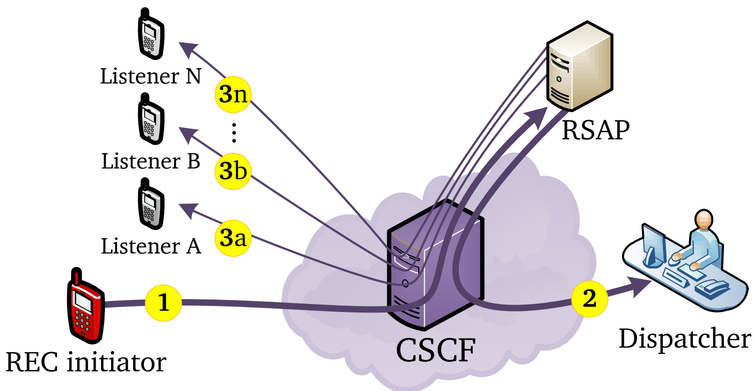


Figure 4.3: Railway Emergency Call (REC) setup procedure in VoLTE (signalling plane). The procedure consists of three steps, during which individual component calls are established.

The REC setup procedure is divided into three steps, during which the individual calls forming a REC are initiated:

1. The REC initiator sends a *SIP INVITE* message to the CSCF with RSAP as the call destination. The CSCF detects that the initiated call is a REC (based on the call destination) and forwards the *INVITE* to the RSAP.

2. Once the RSAP receives the *INVITE*, it initiates a call to the relevant dispatcher. Concurrently, the RSAP continues the call establishment procedure that was initiated in step 1.

3. When the RSAP-Initiator call and the RSAP-Dispatcher call are initiated (but not necessary established yet), the RSAP initiates calls to all other terminals in the relevant railway area. These terminals are referred to as “listeners”, because they only receive the voice and do not transmit anything.

All of the individual calls that form the REC follow the same SIP-based procedure as shown in Figure 4.2. The difference to the one-to-one calls is that REC is answered automatically, so several SIP messages are omitted as redundant, e.g. *RINGING*. Besides, in order to ensure a short setup time and a high availability, every REC receives the highest priority from the underlying LTE network.

Media transmission, i.e. the exchange of packets carrying voice frames, begins when the RSAP-Initiator and the RSAP-Dispatcher calls are established, i.e. steps 1 and 2 are completed. Then, the listeners are added to the REC as soon as their respective call establishment procedures are finished (steps 3a, 3b, . . . , 3n).

The RSAP acts as a media mixer. The REC voice streams between the Initiator and the Dispatcher are routed through the RSAP, which then distributes them among the listeners. This is shown in Figure 4.4.

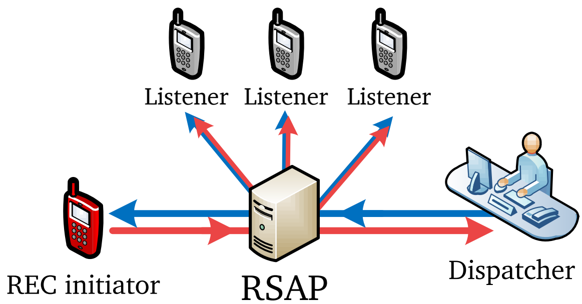


Figure 4.4: Media (voice) flow during the Railway Emergency Call (REC)

4.3 Simulation models and scenarios

4.3.1 VoLTE model in OPNET

In order to validate VoLTE performance, a simulation-based approach was chosen. Models of the two VoLTE calls were prepared in OPNET AppTransaction Xpert (ATX) [59]. Figure 4.5 illustrates the main process of the one-to-one operational call model. Besides the main process, child processes were used for generating the voice traffic, i.e. the media flow.

Figure 4.6 illustrates the main process of the REC model. Also in this case, child processes were used in order to model: the SIP signalling with the listening nodes and the media flow. The implemented models collected various statistics, such as: call setup time, voice packet delay, and voice packet loss. These were used to measure VoLTE performance and to compare it with the railway requirements.

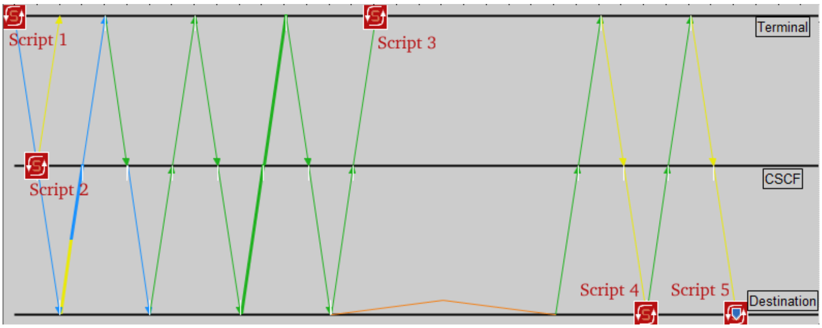


Figure 4.5: VoLTE one-to-one call model developed in OPNET ATX. The main process of the model is shown. Detailed source code of the model is presented in Appendices A.4 and A.5 (pages 163 and 167).

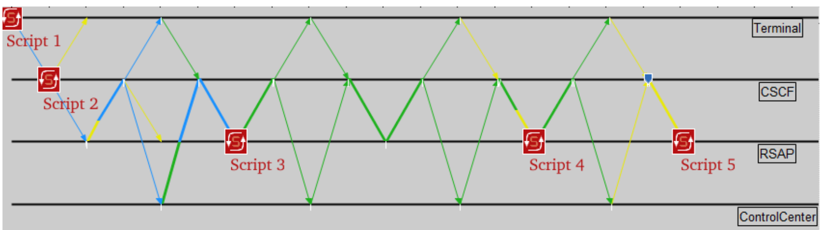


Figure 4.6: VoLTE REC model developed in OPNET ATX. The main process of the model is shown. Detailed source code of the model is presented in Appendices A.6, A.7, and A.8 (pages 171, 175, and 178).

4.3.2 Simulation scenarios

The two VoLTE models were applied in two simulations scenarios modelling LTE networks in typical railway environments. The two scenarios were previously used for ETCS performance validation:

- **Scenario 1 (Snoghøj-Odense line):** The purpose of this scenario was to verify VoLTE performance in a scenario modelling a typical open line railway conditions. Also, this scenario allowed to investigate the impact of alternative radio network deployments on VoLTE performance. There were 10 cases considered with a different number of eNodeBs along the railway line. The number of eNodeBs ranged from 10 to 55. Other details on the scenario were presented previously in Section 3.7.3 on page 55.
- **Scenario 2 (Copenhagen Central Station):** The goal of this scenario was to analyse how the number of voice terminals in a single cell affects VoLTE performance. For that purpose, 6 cases were considered with 5 to 30 trains (UEs) in the cell. The scenario was based on the one presented in Section 3.9.3 on page 75.

Although both scenarios were closely based on those used for ETCS simulations in Chapter 3, they had to be extended: by addition of IMS-specific network nodes, by extending the application mix, and by reconfiguring EPS bearers.

Addition of VoLTE/IMS nodes

For the purpose of VoLTE validation the following three additional nodes were introduced in the network:

- *CSCF node*, which was responsible for modelling the IMS functionality,
- *RSAP node*, which was handling REC signalling and media mixing,
- *Dispatcher node*, which was representing the train dispatcher terminal.

Figure 4.7 shows these three additional nodes in an updated model of the LTE network along the Snoghøj-Odense line. The same nodes were introduced in the model of Copenhagen Central Station.

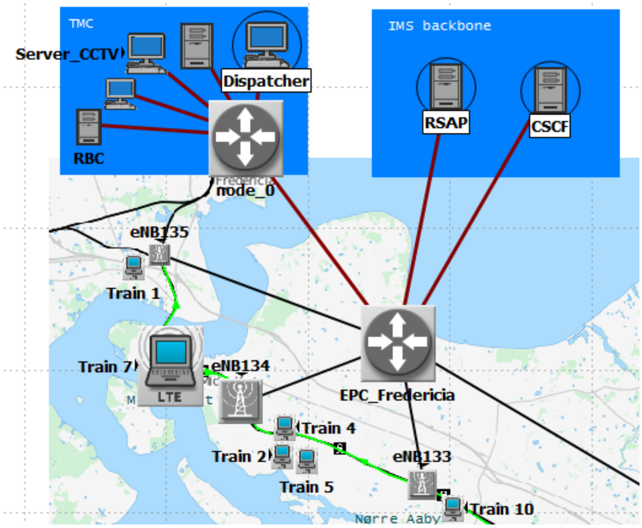


Figure 4.7: Additional VoLTE nodes introduced in the Snoghøj-Odense network model

Extended application mix

The application mix consisted of: one-to-one calls, RECs, ETCS signalling, passenger information, and video surveillance. The non-voice applications were taken from the mix presented in Section 3.9.2 on page 73 (applications 1, 4, 5 and 6). Their purpose was to model a realistic traffic load in the network. The same application mix was used in both scenarios, i.e. Snoghøj-Odense and Copenhagen Central Station. The two VoLTE applications were configured as follows:

- *One-to-one calls*: Every UE was initiating a one-to-one call every 600 s, on average. The called party was randomly chosen between either another UE or the Dispatcher (a fixed terminal). Every call lasted 20 s, on average.

- *REC*: During each simulation run a REC was initiated twice by one of the UEs. An emergency call lasted 60 s, on average.

In both call types, the transmitted voice was coded using AMR codec fixed at 12.2 kbit/s. In each direction (uplink and downlink), an active call was generating 50 voice frames per second. Every frame was 30 bytes long. However, since the frames were transmitted using the Real-time Transport Protocol (RTP), an additional 14-byte RTP header was added to each of them [88, Sec. 5.1].

QoS configuration

Voice communication is one of the critical applications, because it affects railway operation and safety. Due to that, railways require the mobile network to provide appropriate prioritization of the voice communication, namely:

- Voice calls must be prioritized over non-critical applications,
- The various call types must be assigned different priorities depending on the importance of the call [23, p. 21].

Therefore, the modelled LTE network prioritized applications in the following order:

1. REC received the highest priority. Since REC informs the train drivers about a potentially life-threatening situation, the faster it is established, the more time is left for the drivers to take preventive actions. Hence, in the network, REC signalling and REC media transmissions cannot be blocked or delayed by any other traffic.
2. ETCS was the second most important application. It is not the first, because, contrary to REC, the transmission of ETCS does not affect railway safety. Although the ETCS system itself increases the safety, a disturbance in ETCS transmission, even in the worst-case scenario, cannot have severer consequences than forcing the trains to stop. Therefore, during an emergency situation, it is better to risk a disturbance in ETCS transmission than to risk REC being delayed or blocked.
3. One-to-one call received the third highest priority from the network, because it is an important application from the operational point of view, but it does not affect safety.
4. The remaining applications, i.e. the passenger information and the video surveillance, were classified as a best-effort traffic.

This priority order was enforced in the modelled LTE network using EPS bearers. This bearer-based QoS provisioning mechanisms was described before in Section 3.9.2 on page 72. The bearer configuration chosen for VoLTE simulations is presented in Table 4.2.

4.4 Impact of the radio deployment on railway VoLTE (Scenario 1)

This section describes results collected in Scenario 1, which was modelling the Snoghøj-Odense line. In this scenario, there were 10 cases considered. Each

Table 4.2: EPS bearer configuration for VoLTE simulations. Besides the presented dedicated bearers, each UE had a default bearer.

Bearer:	REC signalling	REC media	ETCS	One-to-one signalling	One-to-one media
QCI	1	1	3	2	2
Scheduling priority[1]	2	2	3	4	4
Delay budget[1]	100 ms	100 ms	50 ms	150 ms	150 ms
Packet loss rate[1]	10^{-2}	10^{-2}	10^{-3}	10^{-3}	10^{-3}
GBR Uplink, kbit/s	64	64	32	64	64
GBR Downlink, kbit/s	64	64	32	64	64
ARP[2]	1	2	3	4	5
RLC mode[3]	AM	AM	AM	AM	UM
Transport protocol[4]	TCP	UDP	UDP	TCP	UDP

[1] Pre-configured for a given QoS Class Identifier (QCI), as defined in [75, Tab. 6.1.7]

[2] Allocation and Retention Priority (ARP) is used only by the admission control mechanism during bearer establishment. Afterwards, for scheduling, the priority is pre-configured based on the QCI.

[3] The Radio Link Control (RLC) layer may operate in Transparent (TM), Unacknowledged (UM), or Acknowledged Mode (AM). The difference between these modes is mainly in the data protection mechanisms, which are explained in Section 5.4.1 on page 135.

[4] Transport protocol is configured by the application, not by the EPS bearer. However, since it also has an impact on the QoS, it is shown in the table.

simulation run (each case) lasted 15 minutes and was repeated at least 30 times with varying seed numbers. Presentation and discussion of the results is divided into two sections: the first one presents the call setup time results, while the second presents the voice transmission performance results.

4.4.1 Simulation results: Call setup time

Call setup time is the most important performance indicator according to the railway voice communication requirements [23, pp. 27–28]. The maximum acceptable REC setup time is 2 s. The maximum setup time of a one-to-one call depends whether it is a mobile-to-fixed, a fixed-to-mobile, or a mobile-to-mobile call (see Table 4.1). Since the simulation scenarios consisted of a mix of these calls, the strictest limit of 5 s is used in the following discussions.

Mean call setup time

Firstly, the mean call setup times are investigated, as shown in Figure 4.8. The results are plotted in relation to the number of eNodeBs (radio cells) deployed along the Snoghøj-Odense line. It should be noted that the collected results do not include any time taken by the user to answer the call, i.e. only the delay due to the SIP message transfer and network procedures is taken into account.

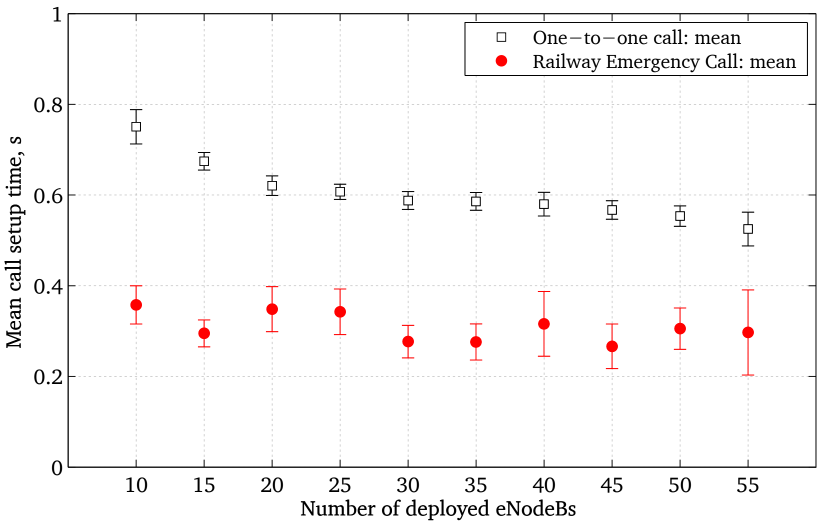


Figure 4.8: Mean call setup time in relation to the number of eNodeBs along the Snoghøj-Odense line (Scenario 1). Error bars indicate 95% confidence intervals.

The first data series in Figure 4.8 presents the results for the one-to-one call. The observed mean setup time was between 0.797 s (the case with 10 eNodeBs) and 0.524 s (the case with 55 eNodeBs). Thus, the more eNodeBs were used for proving the coverage, the shorter was the setup time. This was caused by the following:

- In all of the cases, the total traffic load on the network was the same. However, in a densely deployed network (e.g. the case with 55 eNodeBs), the traffic was distributed over more eNodeBs than in a sparse network. Hence, the traffic load per cell was lower. Since there were only 15 UEs in the modelled scenario, in the cases with a dense deployment, many eNodeBs were only one UE at a time or even no UE at all. Therefore, the radio utilization was very low and the inter-cell interference was minimal. Due to that, queuing and retransmissions on the radio interface were rare. Thus, the message transfer delay and the consequent setup time were relatively lower.

- In a sparsely deployed radio network (e.g. the case with 10 eNodeBs), the average traffic load per cell was relatively high compared to a dense network. Due to the higher traffic load, the Signal-to-Interference-and-Noise Ratio (SINR) was lower, as explained by Salo et al. [66, pp. 6–8]. A consequence of the low SINR was a lower radio throughput. Moreover, the error rate on the radio link increased, so more retransmissions were needed. Each retransmission contributed to the delay of the setup procedure.

Due to the above reasons, the call setup time was longer in the scenarios with a sparsely deployed radio network. Nevertheless, the difference between different deployment strategies is not huge. In all of the cases, the mean setup time was approximately five times shorter than the 5 s requirement. Therefore, the chosen radio deployment should not have any significant impact on the voice users.

Figure 4.8 shows also the mean setup time of the REC. For this call type, the observed values were between 0.254 s and 0.464 s. The confidence intervals are larger in case of REC than in case of the one-to-one call, because only one REC per simulation run was placed. Hence, fewer samples were collected.

Maximum call setup time

Although the mean setup time offered by VoLTE was sufficiently low for the railway purposes, the maximum time had to be also investigated. This is because the call setup time requirements are defined in terms of the maximum acceptable values, not the mean values.

Figure 4.9 shows the maximum setup times recorded in Scenario 1. It can be seen that in case of the one-to-one call, there were few cases when the maximum call setup time exceeded the 5 s limit (which was stricter than the industry limit of 10 s, as presented in Table 4.1). However, these cases above 5 s were very rare—approximately 1.5% of all calls. Therefore, over 95% of calls were established within the 5 s limit, which means that the requirement was fulfilled. In case of REC, the maximum values were below the respective 2 s limit. It means that the call setup time requirement was fulfilled for both call types.

Besides, it should be observed that the call setup time requirements were fulfilled regardless of the chosen deployment strategy. Therefore, although the denser radio network performed better, the eNodeB density did not affect the call setup times significantly.

Furthermore, looking at both Figures 4.8 and 4.9, it is visible that the REC setup was faster that the one-to-one call setup. This observation remained valid in all investigated cases. Despite the longer REC setup procedure, this call was established faster due to the QoS configuration. The EPS bearer that was carrying REC signalling had higher scheduling priority and lower delay budget than the respective one-to-one bearer, as defined by the parameters shown in Table 4.2.

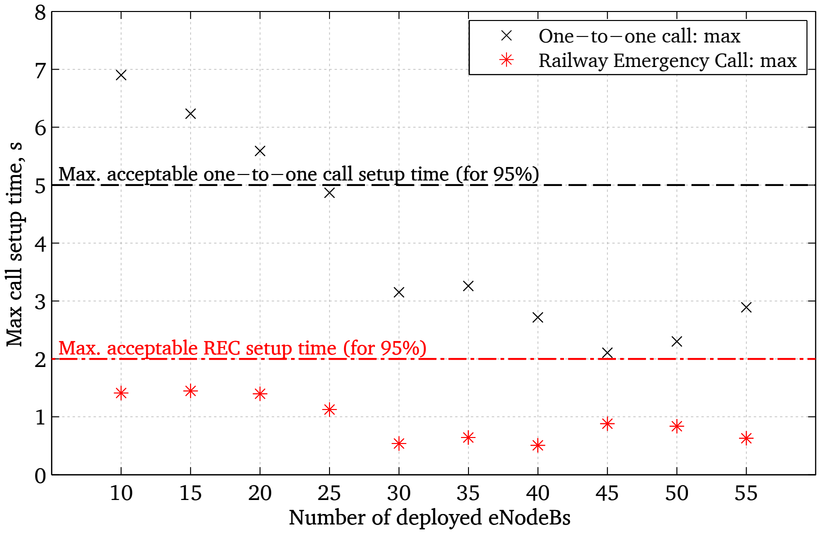


Figure 4.9: Maximum call setup time values recorded in the Snoghøj-Odense scenario. The maximum limits must be fulfilled in 95% of cases.

Therefore REC signalling was prioritized by the packet schedulers and the admission control mechanism.

Another reason for the better REC performance is the exclusive character of this call type, i.e. only one REC could be placed in the network at any time. Hence, in opposite to the one-to-one calls, REC did not compete for network resources with other calls of equal priority.

4.4.2 Simulation results: Voice transmission performance

Besides the setup time in VoLTE, another area of interest was the voice transmission performance in terms of delay and loss of the voice frames. Since each RTP packet carried a single voice frame, the RTP packet transfer delay is equal to the voice transfer delay, while the RTP packet loss rate is equal to the voice frame loss rate.

Voice packet delay

Figure 4.10 shows the mean transfer delay of the RTP packets carrying voice frames during a call. The results are again plotted in relation to the number of eNodeBs deployed along the Snoghøj-Odense line.

Looking at the results concerning the one-to-one call, the mean voice packet delay was between 19 ms (the case with 10 eNodeBs) and 15 ms (the case with

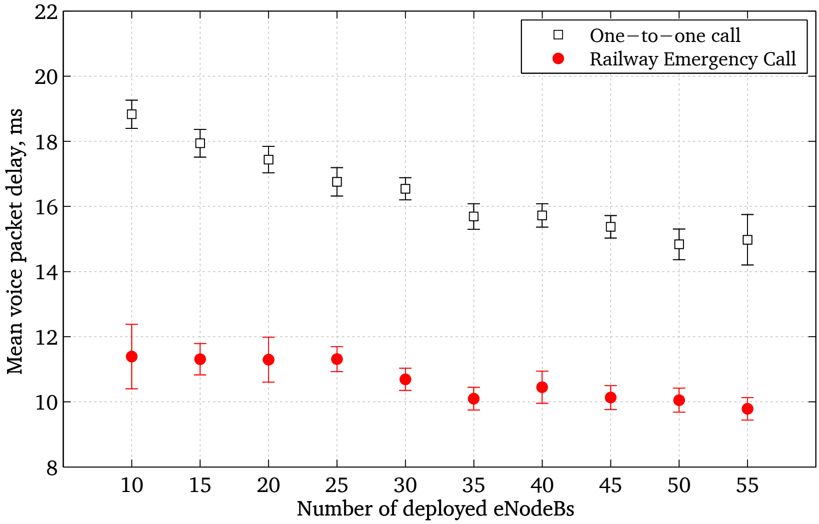


Figure 4.10: Voice packet delay in relation to the number of eNodeBs along the Snoghøj-Odense line (Scenario 1). Error bars indicate 95% confidence intervals.

55 eNodeBs). In case of REC, the mean voice packet delay was between 11 ms (the case with 10 eNodeBs) and 10 ms (the case with 55 eNodeBs). Hence, similarly as with the setup time, the packet delay was lower when more eNodeBs were deployed. The reasons for this tendency were explained in the previous section on the call setup time (see Section 4.4.1 on page 102).

In all of the investigated cases, the observed delays were approximately one order of magnitude smaller than the upper limit of 150 ms.

Voice packet loss

Figure 4.11 shows the mean voice packet loss rate in relation to the number of deployed eNodeBs. In case of the one-to-one call, the observed values were between 1.53% (the case with 10 eNodeBs) and 0.13% (the case with 55 eNodeBs). Since the voice packets were not protected by any retransmission mechanism other than the default Hybrid Automatic Retransmission Request (HARQ), the loss on the radio link resulted in the end-to-end packet loss.

As explained previously in the section on the call setup time, in a sparsely deployed network (e.g. the case with 10 eNodeBs), the probability of a loss on the radio link was higher. Consequently, the voice frame loss was higher as well.

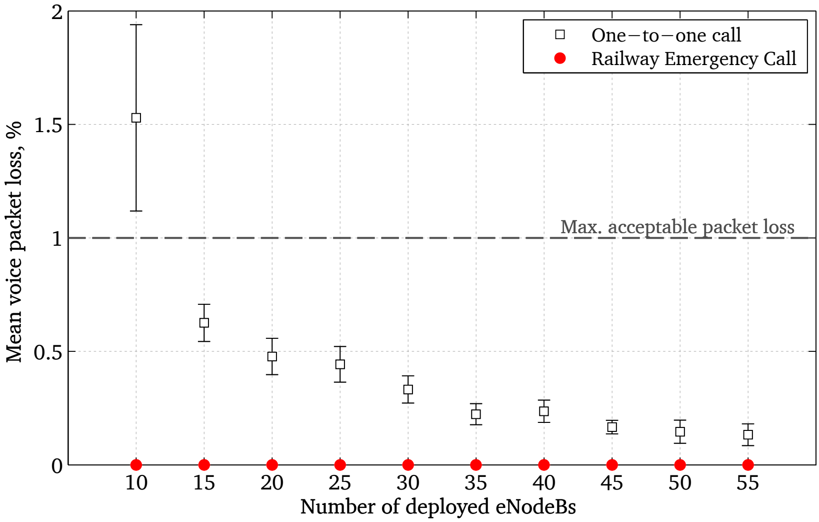


Figure 4.11: Voice packet loss in relation to the number of eNodeBs along the Snoghøj-Odense line (Scenario 1). Error bars indicate 95% confidence intervals.

In the simulation model, it was assumed that voice is coded using AMR. Hence, the maximum acceptable frame loss is approximately 1% [82, p. 5]. This loss requirement was fulfilled in all of the cases, except of the case with 10 eNodeBs.

However, also in this case, the packet loss could be reduced below the 1% limit, if the voice packets were transmitted in Acknowledged Mode (AM) by the RLC layer. AM enables additional retransmission mechanism on the LTE radio interface (*LTE-Uu* interface). The improvement due to the AM is visible in case of the REC, which was transmitted in this mode. Thanks to that, no end-to-end packet loss was observed for this call type. The details on RLC layer and its operational modes is presented in Section 5.4.1 on page 132.

Mean Opinion Score

With the exception of the 10 eNodeB case, the packet delay and the packet loss requirements were fulfilled. However, these two performance-measures describe voice transmission only from the network point of view. For railways, the actual quality of the received voice is more important.

Therefore, an additional voice quality measure should be used, for example, the Mean Opinion Score (MOS), which describes how the call participants would perceive the received voice. MOS is defined in a scale from 1 (the worst quality) to 5 (the best quality). The score is determined by averaging subjective opinions collected in a series of experiments [89]. Hence, this method is not directly applicable to simulation-based approach. However, in this work, MOS values are estimated using *E-model*, as defined by ITU-T [90]. The model allows MOS values to be computed based on the end-to-end voice delay, frame loss, codec, and other parameters. Thus, using E-model, the call quality measured from the network point of view can be “translated” to the user (railway) point of view.

For the *one-to-one call*, in case of 10 eNodeBs, MOS was estimated as 3.8. According to MOS specification [89], values between 3.0 and 4.0 mean that the received voice would be degraded in a “slightly annoying” way. The listener would find the speech understandable with a “moderate effort”. Therefore, the 3.8 MOS result should be seen as acceptable for railway use, but higher values are desired.

In the simulation cases with more eNodeBs, the estimated MOS values were between 4.1 and 4.2. From the listener point of view, the distortion of the received voice would be “audible but not annoying” and “no appreciable effort” would be required to understand the speech. Hence, such a quality should be sufficient for railway operational calls.

Considering *REC*, in all of the cases, the estimated MOS was between 4.2 and 4.3. Therefore, the received speech would be easy to understand, while the voice distortion would not be annoying to the listener. Similarly as in case of the one-to-one call, such quality should be sufficient for railway purposes.

All in all, based on these MOS results, it can be concluded that in all of the investigated radio deployments, VoLTE provided a satisfactory voice quality for both of the call types.

4.5 Impact of the traffic load on railway VoLTE (Scenario 2)

This section describes results collected in Scenario 2, which was modelling Copenhagen Central Station. There were 6 cases considered with a different number of trains UEs at the station. Each simulation run (each case) lasted 15 minutes and was repeated at least 15 times with varying seed numbers. Similarly as in Scenario 1, the discussion of simulation results is divided into two parts that concern call setup time and voice transmission performance, respectively.

4.5.1 Simulation results: Call setup time

Mean call setup time

Figure 4.12 shows the mean call setup time in relation to the number of UEs at the station, i.e. in the LTE cell covering the station. In case of the *one-to-one call*, the mean setup time was between 0.458 s (the case with 5 UEs) and 0.694 s (the case with 30 UEs). In case of *REC*, it was between 0.284 s (the case with 5 UEs) and 0.413 s (the case with 30 UEs).

In the first four of the considered cases—with no more than 20 UEs—the mean call setup time was approximately the same. It began to increase only in the case with 25 UEs, because the traffic load exceeded the available radio capacity. As a consequence, the queuing time on the radio interface increased and the SIP signalling exchange took longer.

It should be noted that the jamming nodes, which were modelling two neighbouring cells (see Figure 3.23 on page 76), were transmitting at a constant rate in all of the cases. Therefore, in Scenario 2, the inter-cell interference from the point of view of the central cell was at the same level in all of the cases. In opposite to that, in the previously discussed Scenario 1 (Snoghøj-Odense line) the inter-cell interference varied from case to case.

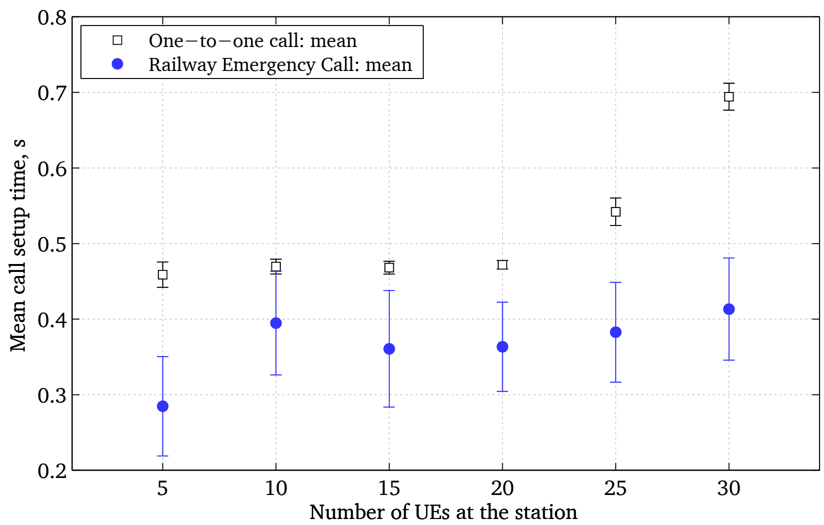


Figure 4.12: Mean call setup time in relation to the number of UEs in the LTE cell at Copenhagen Central Station (Scenario 2). Error bars indicate 95% confidence intervals.

For both call types, the more UEs were at the station, the longer it took to establish a call. Nevertheless, the mean setup time was approximately five times shorter than the maximum 2 s and 5 s limits defined in the railway requirements. Therefore, regardless of the traffic load in the cell, VoLTE provides fast call setup that fulfils the railway requirements in realistic scenarios.

Maximum call setup time

Figure 4.13 shows the maximum call setup time results observed in Scenario 2. The longest setup time was recorded in the case with 30 UEs at the station. In that case, it took a maximum of 1.63 s to establish a one-to-one call and 0.71 s to establish a REC. As visible in the figure, both of these values are significantly lower than the maximum 2 s and 5 s limits defined by the railway industry.

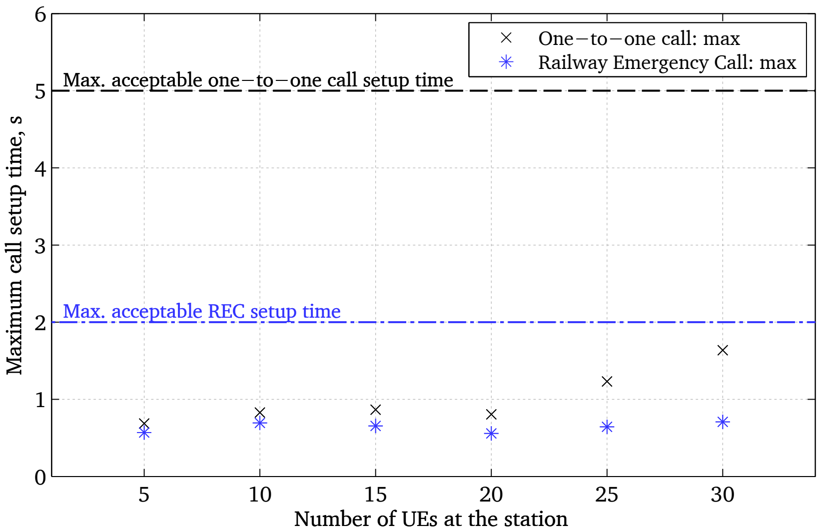


Figure 4.13: Maximum call setup time values recorded in the Copenhagen Central Station scenario

4.5.2 Simulation results: Voice transmission performance

Voice packet delay

Figure 4.14 shows the mean transfer delay of the RTP packets delivering voice frames during a VoLTE call. Considering the one-to-one call, the mean delay was between 12 and 13 ms. Considering the REC, the delay was approximately 8 ms.

Similarly as in Scenario 1, the observed values were at least an order of magnitude smaller than the limit of 150 ms.

These delay results remained stable regardless of the number of UEs at the station. This is because, the voice packets were transmitted over the dedicated EPS bearers, which guaranteed a minimum bitrate that was higher than the bitrate of a voice stream. Moreover, the QoS configuration prioritized the voice packets over the non-critical applications, which constituted the main portion of the traffic load.

It is worth noting that the voice packet delay did not increase when the network was overloaded (the cases with 25 and 30 UEs), while the call setup time did, as was shown in Figure 4.12. This difference is a result of the difference in the length of the signalling and the media packets. The SIP signalling messages were approximately 400–1200 bytes long, while the voice packets were 44 bytes long (without taking into account an overhead from the transport and the IP layers).

Both the media flow and the signalling flows were guaranteed a minimum bitrate of 64 kbit/s. In the uncongested cases—with 20 UEs or fewer—the network had available resource to offer a higher bitrate than the guaranteed one. However, once the network became congested, it could guarantee only the 64 kbit/s, as defined in the EPS bearer configuration. This bitrate reduction did not affect the media flow, because the voice packets were relatively small. On the other hand, it affected the SIP signalling packets, which were much longer and had to queue on the radio interface.

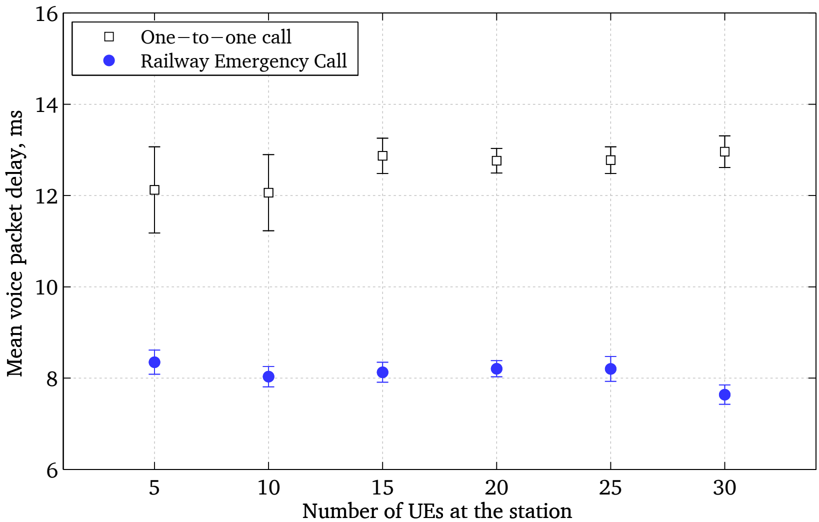


Figure 4.14: Mean voice packet delay in relation to the number of UEs in the LTE cell at Copenhagen Central Station (Scenario 2). Error bars indicate 95% confidence intervals.

Voice packet loss

Figure 4.15 shows the results concerning the RTP packet loss, i.e. voice frame loss. Considering the one-to-one call, the mean packet loss rate was between 0.000% and 0.001% depending on the simulated case.

Since the confidence intervals are larger than the recorded values, the collected samples are not sufficient to draw conclusions about the relation between the loss and the number of UEs. Nevertheless, such a small packet loss, which is three orders of magnitude smaller than the maximum acceptable 1%, would not have any noticeable impact on the quality of the received voice.

Considering the REC, no packet loss was observed, because the voice packets were protected with the retransmission mechanism on the RLC layer.

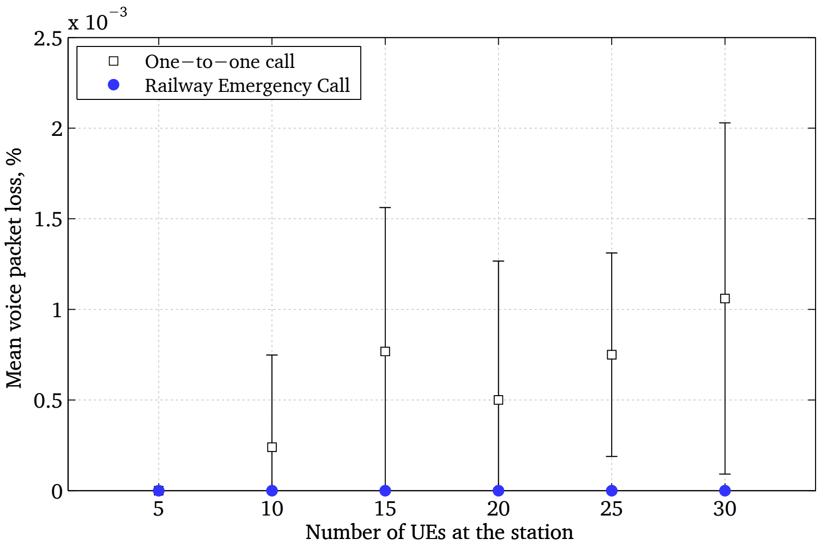


Figure 4.15: Voice packet loss in relation to the number of UEs in the LTE cell at Copenhagen Central Station (Scenario 2). Error bars indicate 95% confidence intervals.

Mean Opinion Score

Based on the packet delay and the packet loss results, MOS values were estimated, similarly as in Scenario 1. In all of the cases, for both call types, MOS was approximately 4.2. Thus, the received voice should have good quality with minor disturbances. This means that despite the growing traffic load on the cell, the modelled VoLTE network offered a good voice quality.

4.6 Chapter conclusions

Voice communication is an essential application, which is used in everyday railway operations and, also, in case of emergency situations. From the railway point of view, its importance is comparable with the importance of ETCS. Therefore, the future railway communication network must provide voice communication that fulfils the specific railway requirements.

LTE may become the future railway mobile technology, only if it can provide railway voice communication fulfilling all these requirements. Therefore, in this chapter, VoLTE was analysed as a possible solution for providing railway voice communication in the future. Two VoLTE models—of the one-to-one operational call and of the Railway Emergency Call—were developed. Using these models VoLTE performance was validated in two simulation scenarios modelling typical railway scenarios.

Simulation results showed that VoLTE provides a fast call setup procedure that allows calls to be established significantly faster than required by railways. This is especially important in case of the REC, which is very important and highly requested by railways. Since the REC setup time in VoLTE is noticeably better than required by railways, the REC can fulfil its goal of providing railway personnel with a nearly instant warning about an emergency situation. What is important, the fast setup can be achieved regardless of the chosen radio deployment strategy and regardless of the traffic load in the network.

Moreover, the simulations showed that VoLTE offers voice transmission with very good performance in terms of packet transfer delay and satisfactory performance in terms of packet loss. Therefore, the received voice quality in a VoLTE call should be sufficiently high for railway purposes.

Finally, the simulation results demonstrate that the bearer-based QoS mechanism of LTE successfully differentiates between the two call types. It also prioritizes the voice communication over background network traffic. Even under unfavourable conditions, e.g. traffic overload, the one-to-one call, and the REC were established within the time limits and the received voice quality was good. Therefore, the bearer-based mechanism is able to provide the necessary prioritization and protection of the critical communication, as requested by railways.

All in all, based on the results, it can be concluded that VoLTE is able to provide voice communication fulfilling the railway requirements in terms of features and performance. Hence, VoLTE should be considered as a valid candidate to replace GSM-R as the technology providing operational and emergency voice communication for railways.

CHAPTER 5

Heterogeneous radio networks for railways

Communication networks became vital elements of modern railways. They provide critical applications, such as ETCS and voice communication. Thus, indirectly, the communication networks make railways safer, more efficient, and more competitive compared to other means of transport.

However, the bigger is the role of communication-based applications, the bigger is the impact of the communication problems on railway operations. For instance, if ETCS messages are obstructed due to an insufficient communication capacity or due to a communication failure, trains may be unable to run. Therefore, the reliability of a communication system may be equally important as the reliability of the more classical railway elements, such as interlockings and track switches.

A reliable communication system is be characterized by the following: resiliency against various failures, sufficient transmission capacity, and integrity of the transmitted data. Only such a system can offer the necessary high availability of the railway applications.

In order to improve network resilience, railway mobile networks are often deployed with a great amount of redundancy. This approach reduces the impact of various hardware and software failures. However, the expensive redundant deployment may offer additional benefits besides the high resilience. Therefore, in this chapter, a new concept of a heterogeneous railway radio access network is introduced. Its purpose is not only to improve communication resilience and robustness, but also to offer additional network capacity and optimize the radio deployment for different types of railway users. The additional capacity is especially important in the places where high-density train traffic is expected, e.g. major train stations. In such places, a traditional radio deployment based on large macro cells may struggle to offer sufficient capacity for bandwidth-demanding applications. Besides, the proposed architecture can also facilitate the migration to a new wireless technology that may succeed GSM-R in the future.

Apart from the resilience and the capacity, another element of the reliable communication is the data integrity, i.e. prevention of data losses in the network. This aspect is especially important for ETCS signalling and other critical applications. Therefore, in this chapter various data protection mechanisms are discussed and validated on an example of LTE. Their performance is compared with ETCS requirements.

Chapter organization

This chapter is organized as follows. Section 5.1 describes the radio network deployments that are typically used by railways. Section 5.2 proposes a new radio deployment architecture with potential advantages in terms of capacity and radio utilization. In Section 5.3 this new architecture is validated in simulation scenarios. Finally, Section 5.4 discusses mechanisms for providing ETCS data integrity, despite unreliable physical transmission.

5.1 Typical railway radio access deployments

Communication in a mobile network, such as GSM-R or LTE, is sensitive to various internal and external factors, such as:

- Hardware and software failures of radio base stations,
- Severe weather conditions that damage the vulnerable networks elements, e.g. base stations and antennas,
- Backbone failures due to node breakdowns or cable/fiber cuts,
- Power supply failures,
- Wireless transmission failures due to external interference or purposeful electromagnetic attacks.

Even the most careful hardware and software design cannot entirely eliminate the risk of the above failures. This is due to their random nature. Failures are often difficult or even impossible to predict.

At the same time, all of these failures may have significant consequences, because they may break all communication between the train On-board Unit (OBU) and the Radio Block Centre (RBC). As it was explained in Section 3.1 on page 34, without the OBU-RBC link, ETCS signalling cannot work and trains cannot drive. Moreover, since the voice communication is provided over the same network as ETCS, it may be also interrupted or blocked by the same network failure. Therefore, in such a worst-case scenario, train drivers are cut off from the two possible ways of receiving commands from a dispatcher: signalling-based and voice-based.

In railway mobile networks, the inevitable risk of failures is often reduced via a *redundant network architectures* [12, p. 164]. Since network elements may fail, they are doubled by installing backup elements, which take over when the original element breaks down. In a redundant network, the resilience against failures is increased. Thus, the network and application availability is also increased.

The concept of redundancy can be applied in various ways, which usually provide different balance between a network cost and its availability. The more redundant the network, the more expensive it is, but on the other hand, it offers a higher availability.

One of the redundant approaches is to deploy two radio base stations (two transceivers) at each radio mast. This design is illustrated in Figure 5.1. If the main base station fails, then the backup base station is used to provide the connectivity with the trains (OBUs). Thus, this network architecture protects against hardware and software failures of a base station. Moreover, if the redundant base station operates at different frequency, the network may be also protected against interference and electromagnetic attacks—to some extent [91].

This approach to radio network design is used, for example, in the Norwegian GSM-R network [58, p. 16]. Additionally, all Norwegian base stations are equipped

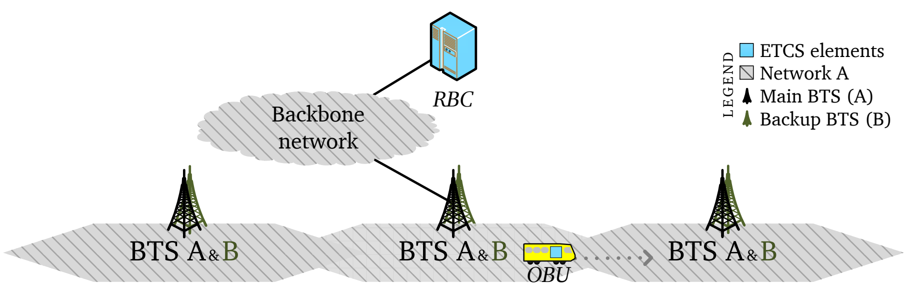


Figure 5.1: Radio access network with redundant base stations

with an independent eight-hour backup power supply, which minimizes the impact of power failures on the network operation.

Another redundant approach is to deploy two independent radio access networks, as illustrated in Figure 5.2. In this case, the main “Network A” is supported by a redundant “Network B”. The base stations of both networks are deployed in a dovetail fashion, i.e. a base station B is deployed in the middle between two neighbouring base stations A. At any point along the railway line, there is coverage from at least two base stations. Since the base stations are not collocated, the risk that both of them are damaged due to some external force (e.g. lightning, falling tree, construction works) is greatly reduced. Additionally, the dovetail deployment increases the signal coverage, because in the locations where the signal of Network A is weak, the signal of Network B is strong. This approach to radio network design is used in the Swedish GSM-R network [57, p. 16].

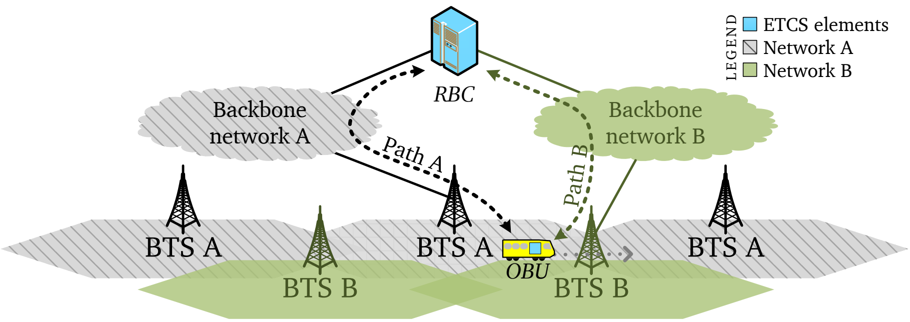


Figure 5.2: Radio access network with double coverage along a railway line. Two disjoint paths are provided between the OBU and the RBC. The two networks provide coverage over the same area. However, for the purpose of clarity of the figure, Network B is shifted vertically.

Furthermore, in the example in Figure 5.2, there is also redundancy in the backbone part. Base stations A and B have their respective independent backbone networks. This means that there are always two disjoint communication paths between the OBU and the RBC. Therefore, this architecture protects against failures in both the radio and the backbone network parts.

In Denmark, the GSM-R network is deployed either with a single or with a double (redundant) coverage depending on the area [19, p. 6]. The double coverage is provided where the high intensity of train traffic is expected. In these places, a communication failure could lead to significant operational disruptions. Thus, the network availability is especially important there.

5.2 Heterogeneous macro/micro radio network

The purpose of these typical redundant network architectures, which were described in the previous section, is to provide high network availability. Railways gain little benefit from the redundant network, as long as the main network operates correctly.

The aim of the research work that is presented in this chapter was to propose a network architecture that would bring additional benefits besides the high availability. Therefore, an alternative heterogeneous radio access network is proposed, as illustrated in Figure 5.3. This double coverage architecture is referred to as "macro/micro" in the following sections.

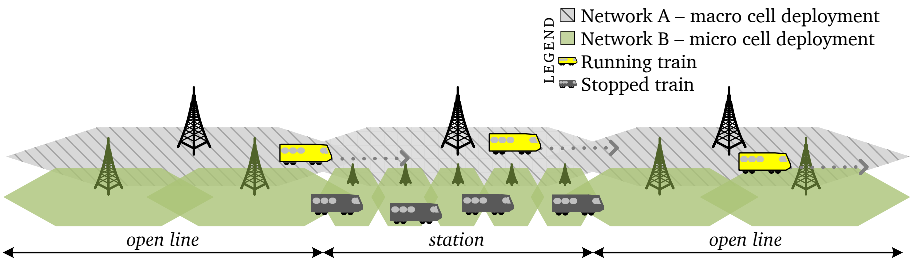


Figure 5.3: Proposed "macro/micro" heterogeneous radio network architecture with two radio levels

The proposed network architecture is composed of two radio levels: macro and micro. The *macro* level consists of radio cells with a relatively long radius, i.e. measured in kilometres. A single macro cell can cover a whole station or a long section of an open railway line. Thus, the macro level is similar to the GSM-R deployments used today.

The *micro* level consists of considerably smaller cells with a shorter radius, i.e. measured in meters. At the micro level, the cell radius varies more than at the macro level. In the high-destiny areas, where intensive train traffic is expected, e.g. at train stations, the micro cells have a short radius. On the open line, where the capacity demand is lower, the micro cells may be bigger and their size may approach the macro cell size.

The proposed macro/micro network architecture, which is novel in railways, is expected to bring a number of advantages as described in the following paragraphs.

High network availability

The proposed network architecture provides full double coverage over the entire railway area. At any point along the tracks, a train receives radio signals from at least two base stations. Therefore, the same level of network availability is offered as in the classical double coverage network that was illustrated in Figure 5.2.

Optimized cell deployment for different users

In a mobile cellular network, the more radio cells are deployed (i.e. the smaller is the cell radius), the higher is the total radio capacity. On the other hand, the bigger are the cells, the fewer handovers must be performed by a moving terminal. Thus, the classical radio network deployment must balance, i.e. compromise, between the capacity and the handover rate.

The proposed macro/micro architecture can achieve both: the high radio capacity and the low handover rate. At the macro level, the handover rate is low, because the cells are large. Therefore, in normal operation when both levels work without failures, the macro level is intended to be used by the running trains.

At the micro level, the radio capacity is high, because there are many cells deployed. Hence, this level is intended for the stopped trains and the slow-moving handheld terminals. Due to their low speed, the inter-cell handovers are not an issue for these terminals.

Introduction of high-frequency radio bands

Railway mobile networks must provide coverage over long rail-roads in order to serve high-speed users. Due to these specifics, the railway networks are usually based on a relatively small number of long-radius cells. Thanks to this approach, the radio deployment is cheaper (small number of base stations) and the handover rate is minimized.

Due to the signal path loss phenomena, a radio cell operating at a high carrier frequency has much shorter radius than a one operating at a lower frequency (see Figure 3.12 on page 54). Hence, only low-frequency bands, such as the 900 MHz GSM-R European band, are usually considered for railways. This radio band is at low frequency, but it is only 4 MHz or 7 MHz wide [4, p. 148]. Regardless of the chosen mobile technology, this narrow bandwidth is one of the most important factors limiting the radio capacity (throughput).

Since there is a high demand for the low-frequency bands, railways may not receive any additional bandwidth at these attractive frequencies [11, p. 35]. However, the proposed macro/micro architecture opens a possibility of using also the high frequencies, which traditionally were not considered for railways. The micro-cells cover much smaller areas. As a consequence, they may operate at a higher carrier frequencies, such as the 5.9 GHz band that is allocated for the Intelligent Transportation Systems (ITS) [92].

The second frequency band increases the bandwidth and, in turn, the available radio capacity. Besides, it has also an additional advantage. The second frequency band can be considered as a “frequency redundancy”. If one of the bands is interfered (purposefully of accidentally), all communication may be handed over to the other band. Thus, the second band increases network resilience.

Increased network capacity

The proposed macro/micro architecture should offer a considerably higher capacity than the classical macro-cell-only deployment. There are three reasons for that. Firstly, because there is a significantly higher number of radio cells introduced at the micro level. Secondly, because the micro-cells can utilize the high-frequency bands, which were unavailable to railways before. Thirdly, because the shorter is the cell range the better is the cell edge throughput [66, p. 8].

In the classical deployment, utilization of the radio cells varies depending on the location. At train stations, it is much higher than along open lines. This is due to the difference in the concentration of trains at these locations. While passing through a station, a train (terminal) passes through a highly utilized radio cell. On the other hand, the trains stopped at the station experience a sudden traffic peak from that passing train. Thus, such a single-level network has a high variance of the traffic load. This may cause disturbance for the non-critical applications, which may need to be pre-empted due to the temporary traffic peak.

Contrary to that, in the proposed macro/micro deployment, the traffic load is more stable. This is because, the running and the stopped trains are separated at the two different radio levels. Thus, a running train should experience stable traffic load in all of the cells it traverses, while the stopped trains should not experience a traffic peak due to a passing train.

Multi-technology networks

The two radio levels can be based on the same or on different radio access technologies. For instance, the macro-level may use LTE. The micro-level, on the other hand, may use technologies based on IEEE 802.11 (Wi-Fi), which have not been used in the mainline railways, due to their relatively short range. However, these technologies are popular in CBTC urban railway systems [6], while mainline railways express their interest in using them [33, p. 47].

Another option could be to treat the current GSM-R network as the macro level and deploy LTE (or other future railway communication technology) at the micro level. In this way, the proposed architecture can serve as a transition strategy for migration to a new wireless technology. This transition phase is expected to take many years [33, p. 30]. Thus, it is important to make the migration smooth and gradual. Thanks to the heterogeneous architecture, compatibility with the non-upgraded terminals would be maintained, because they would use the GSM-R-based macro-level. On the other hand, the upgraded trains could already benefit from the new LTE-based micro level.

Such a setup with coexisting LTE and GSM-R should be particularly advantageous, because the EPC, i.e. the LTE backbone, provides standardized mechanisms for interoperability with the legacy GSM networks [25, pp. 254–258].

5.3 Capacity gain in the micro radio deployment

In comparison to a classical single-level architecture, one of the main advantages of the proposed macro/micro architecture is the additional radio transmission capacity. Even if a high-throughput wireless technology is used (e.g. LTE), the additional capacity is important, because of the increasing demand for non-critical railway applications, which are often more bandwidth-demanding that currently used applications.

The simulation results that were presented in Chapters 3 and 4 demonstrated that LTE offers a significantly higher capacity than GSM-R. Even in the small 5 MHz bandwidth, LTE offers enough capacity to provide the critical railway applications, i.e. ETCS and voice communication. However, besides the critical applications, railway communication networks provide also non-critical applications. The demand for them is already increasing [11, p. 40] and it is expected to increase more in the future [9, pp. 38–39]. As presented in Section 3.9.2 on page 72, examples of such non-critical applications are: tele-maintenance, video surveillance, platform surveillance and Internet access for passengers. Many of these applications require considerably more network bandwidth than ETCS and voice communication. Even the capacity offered by LTE may be insufficient to provide these bandwidth-demanding applications in high-density railway areas, such as major train stations.

The two bottlenecks that limit the network capacity are the narrow railway radio spectrum and the macro-cell based deployment. The macro/micro architecture may be able to address these problems and, as a consequence, may significantly increase the radio capacity. Therefore, it was chosen to investigate this new architecture in computer-based simulations. The specific goals of these simulations were as follows:

- Analyse the capacity increase offered by the additional micro-cell radio level in comparison to the macro-cell level on an example of LTE railway communication network. Although in this research work an LTE network is considered, the macro/micro architecture could be applied to other mobile technologies.

- Investigate the impact of the micro-cell level on the performance of exemplary railway applications, namely: ETCS signalling, voice communication, and video surveillance. The performance is measured in terms of end-to-end transfer delay and packet loss.

- Consider the micro-cell deployment under varying train traffic intensity and the consequent communication traffic load.

The research work presented in this section has been previously published in a paper [Sniady2014a].

5.3.1 Simulation scenarios

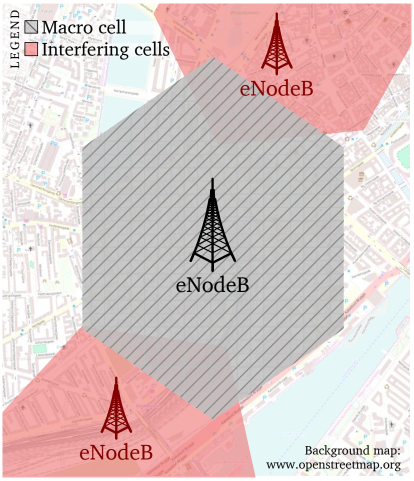
For the simulation purpose, it was chosen to model an LTE mobile network covering Copenhagen Central Station. As presented in Section 3.9.1, this is the station area with the highest concentration of trains in Denmark [71, p. 11]. Therefore, it is also the area where the railway mobile communication network must provide the highest capacity.

Two LTE deployment scenarios were considered in the simulations: a macro-cell based and a micro-cell based. The macro/micro architecture is a combination of these deployments.

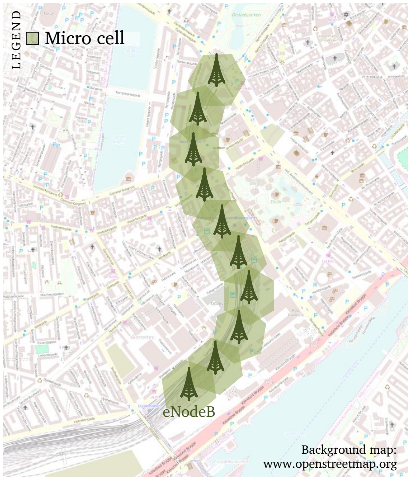
In the **first scenario**, which is illustrated in Figure 5.4a, the station was covered with a single LTE *macro-cell* (i.e. a single eNodeB) with a radius of approximately 1 km. The cell operated in a 5 MHz spectrum, in the 900 MHz band assigned currently to GSM-R (see Figure 2.4 on page 17). Since LTE is an interference-limited technology, four jammer nodes were deployed besides the central macro cell. Their role was to model a realistic level of interference that can be expected from the neighbouring cells.

In the **second scenario**, which is illustrated in Figure 5.4b, the station was covered with 10 *micro-cells*, each having an approximate radius of 50 m. The micro-cells were deployed in a linear fashion, following the shape of the station tracks. Also in this scenario, the network operated in a 5 MHz spectrum, but at the 5.9 GHz frequency band. In this scenario, jammer nodes were not used, because the micro cells themselves introduced the inter-cell interference.

Both deployments ensured a minimum −92 dBm signal power in the whole station area, as required by ETCS (see Section 3.4 on page 40). Further details on the parameters used in the simulations are shown in Table 5.1.



(a) Macro-cell deployment



(b) Micro-cell deployment

Figure 5.4: Two radio network deployments at Copenhagen Central Station considered in the simulations

Table 5.1: Simulation parameters and configuration

Parameter	Macro	Micro
Carrier frequency[1]	920 MHz	5.9 GHz
Bandwidth	5 MHz	
eNodeB antenna height[2]	50 m	10 m
eNodeB antenna gain[3]	15 dBi	
UE antenna height[4]	4 m	
UE antenna gain[3]	1 dBi	
Path loss model[5]	Urban Macro (UMa)	Urban Micro (UMi)
Multipath channel model[6]	ITU Pedestrian A	

[1] The macro-cell operated in the band used today by GSM-R: 876 to 880 MHz band in the uplink and 921 to 925 MHz band in the downlink. The micro-cells operated in the 5.9 GHz band assigned for the Intelligent Transportation Systems (ITS) [92].

[2] Assuming that the macro eNodeB antenna is placed on a mast, while the micro eNodeBs are attached to the station's ceiling.

[3] Chosen within the typical range as given in [39, p. 223].

[4] Assuming that the UE antenna is placed on a train roof [22, p. 41].

[5] ITU-R M2135 path loss models were chosen, because they are applicable to densely urbanized areas, such as the one around Copenhagen Central Station. Moreover, UMi supports the 5.9 GHz carrier frequency [93].

[6] Since the trains had fixed positions, low-speed channel model was used.

5.3.2 Application mix and QoS configuration

For the simulation purpose, an application mix was prepared, similar to the one described in Section 3.9.2 on page 72. It consisted of five critical and non-critical applications:

1. *The European Train Control System (ETCS)*, as described in detail in Chapter 3, was based on a message exchange between the OBU and the RBC. Each OBU was sending a 128-byte message to the RBC every 30 s, on average. The RBC replied to the OBU with a 128-byte message. The inter-message time interval was based on assumptions given in an example published in "ETCS for Engineers" handbook [4, p. 155]. The message length was chosen as defined in ETCS requirements [19, p. 18]. Moreover, according to these requirements, the mean ETCS transfer delay must be below 500 ms, while the probability of data loss cannot exceed 0.01%.

2. *Voice communication (telephony)*, as described in detail in Chapter 4, was used for driver-dispatcher communication. In this scenario, each driver was making a one-to-one call to the dispatcher (represented by an application server in the model) every 900 s, on average. Each call lasted for 20 s, on average. It generated one downlink and one uplink 64 kbit/s voice streams. Voice call requires mean "mouth-to-ear" delay below 150 ms and packet loss below 1% (see Section 4.1.2 on page 89).

3. *Voice announcements* were informing the on-board passengers about delays and other changes to the travel schedule. An announcement was sent to each train every 900 s, on average. Each announcement lasted 5 s and generated a 64 kbit/s downlink stream.

4. *Video surveillance* was transmitting two real-time video streams from on-board cameras to a train security centre (represented by an application server). Each of two cameras generated a 500 kbit/s uplink stream. Maximum acceptable mean transfer delay was 150 ms, while the maximum packet loss was 1% [94, Sec. 5.5, Table 1].

5. *Tele-maintenance* was delivering a data that was collected by on-board sensors to the maintenance centre (represented by an application server). Every 20 hours, on average, a 7 GB file was uploaded from each train.

QoS configuration

Table 5.2 shows the EPS bearer configuration used in the simulations. The purpose of the chosen configuration was to ensure prioritization of the critical railways applications, i.e. ETCS signalling and voice communication. Therefore, these two

Table 5.2: EPS bearer configuration for macro/micro simulations

Bearer:	**ETCS**	**Voice**	**Default**
QCI	3	2	9
Scheduling priority*	3	4	9
Delay budget*	50 ms	150 ms	300 ms
Packet loss rate*	10^{-3}	10^{-3}	10^{-6}
GBR Uplink	16 kbit/s	64 kbit/s	—
GBR Downlink	16 kbit/s	64 kbit/s	—
ARP	1	5	9

* Pre-configured for a given QCI, as defined in [75, Tab. 6.1.7]

applications received dedicated EPS bearers. The remaining applications were delivered over the default bearer.

5.3.3 Simulation results

The simulation goal was to investigate the capacity difference between the two radio deployments. Each of the deployments (scenarios) was considered in 10 cases, which differed in the number of trains (UEs) at the station: from 5 to 50 UEs.

Each simulation run lasted 20 minutes and was executed 15 times with different seed numbers. The following sections present the collected results concerning: radio throughput, ETCS signalling, the voice communication and the video surveillance. From the whole application mix, ETCS and the voice communication were chosen because these are critical railway applications. The video surveillance was chosen as an example of a bandwidth-demanding application.

Radio throughput

Firstly, the two deployments are compared in terms of the offered radio capacity. For this purpose, Figure 5.5 shows the mean uplink radio throughput in relation to the number of trains (UEs) at the station. The presented throughput is a sum of the throughput from all eNodeBs at the station.

The traffic load in the uplink direction (from a train to a server) was significantly higher than in the downlink. This was due to the video surveillance and the tele-maintenance applications, which both sent a significant amount of data in the uplink. Therefore, the uplink results illustrate better the difference in the available radio capacity.

In both scenarios (deployments), when 5 trains were placed at the station, the mean uplink throughput was approximately 5 Mbit/s. When the number of

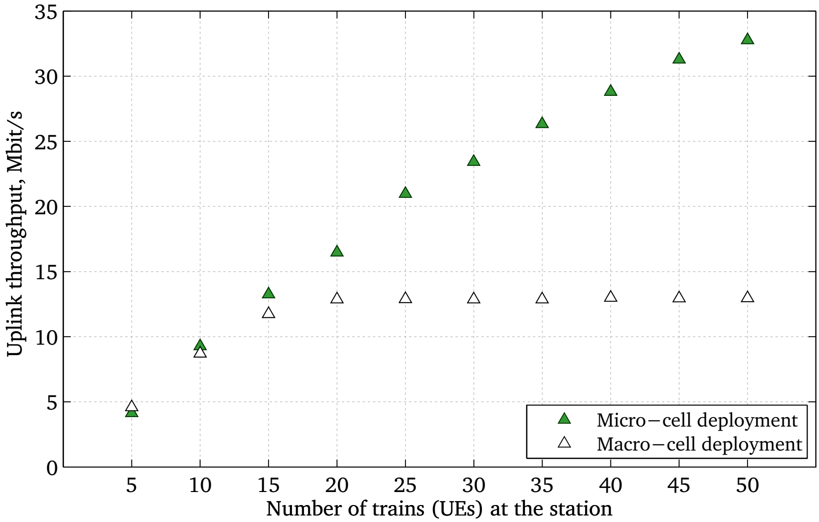


Figure 5.5: Mean uplink radio throughput from all eNodeBs in relation to the number of trains at the modelled station. Two deployments were considered: macro-cell based and micro-cell based.

trains increased to 10, the throughput also increased proportionally, to approximately 10 Mbit/s. However, in the following cases with more trains, the two radio deployments behaved differently.

In the *macro-cell* deployment, the throughput reached its maximum value of 12.9 Mbit/s already in the case with 20 trains. When more trains were introduced in the model, the throughput did not increase. It means that the traffic load exceeded the available radio capacity.

In the *micro-cell* deployment, the uplink throughput continued to increase as more trains were introduced in the model. In the case with 50 trains, it reached 32.8 Mbit/s. However, even in this case, the micro-cell radio throughput did not reach saturation. Thus, the radio capacity was larger than the traffic load.

Comparing the two result series in Figure 5.5, it is visible that the micro-cell deployment offered a significantly higher radio capacity than the macro-cell deployment. This difference was a result of the greater number of cells in the micro-cell deployment. Due to that, the same traffic load, instead of being served by one cell, was distributed over 10 micro-cells.

The following sections describe how this difference in available radio capacity affected the particular railway applications.

ETCS signalling

Performance of ETCS transmission was measured in terms of the mean end-to-end transfer delay and the mean packet loss.

Figure 5.6 shows the mean ETCS transfer delay measured between the train OBU and the RBC. In the case with 5 trains at the station, the delay was approximately 14 ms in both deployments. In the *macro-cell* based deployment, as more trains were introduced in the model, ETCS delay grew more rapidly due to the higher traffic load per cell. It reached almost 40 ms in the case with 20 trains.

The delay did not grow higher in the following cases due to the QoS mechanism. The LTE scheduler, which is responsible for fulfilling the QoS targets for each EPS bearer, was configured to prioritize those bearers which reached 80% of their delay budget. As was shown in Table 5.2 on page 124, the bearer carrying ETCS traffic had a delay budget of 50 ms. Thus, when ETCS messages queuing time on the radio interface approached $80\% \cdot 50\ ms = 40\ ms$, the ETCS bearer was given a greater weight by the scheduler. In this way, the mean delay was kept within the budget.

In the *micro-cell* deployment, ETCS transfer delay grew noticeably slower than in the macro-cell deployment. Only in the case with 50 trains, the results from the two deployments approached similar values. The micro-cell deployment performed better, because the traffic load was distributed over a larger number of radio cells. As a consequence, the radio utilization was lower. Especially important for keeping

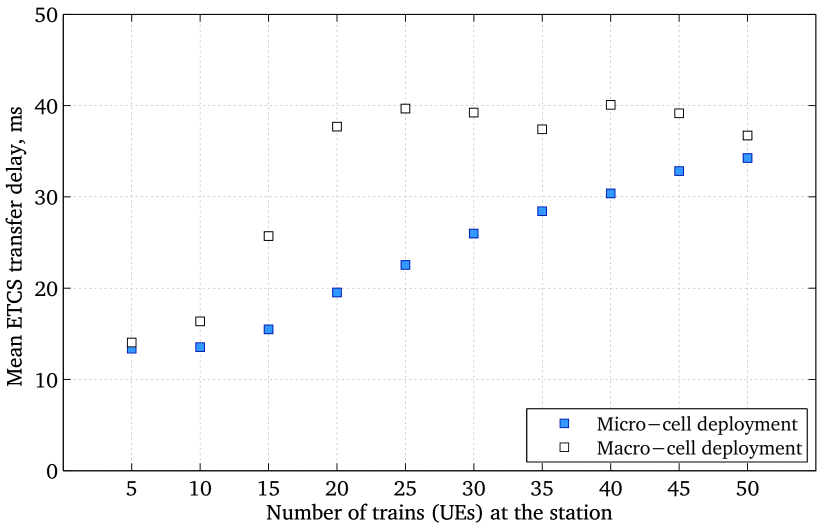


Figure 5.6: ETCS end-to-end transfer delay in the two alternative deployments considered for Copenhagen Central Station.

delay low, was the lower utilization of the control channels, namely the Physical Uplink Control CHannel (PUCCH) and the Physical Downlink Control CHannel (PDCCH). Overload of these channels is one of the source of delay in LTE [77].

Despite the differences between the scenarios, in both of them, ETCS delay was at least an order of magnitude below the maximum acceptable 500 ms. Thus, both deployments fulfilled ETCS requirement on the transfer delay.

Figure 5.7 shows the mean ETCS packet loss in relation to the number of trains. The observed values were in a range between 0.04% and 1.0%. In both scenarios, the packet loss was considerably higher than the maximum acceptable 0.01% [19, p. 18]. Hence, the requirement was not fulfilled.

The reason for this high packet loss was the fact that no retransmission mechanism was used except the default Hybrid Automatic Retransmission Request (HARQ) on the radio link. Any ETCS packet loss in the network resulted in an irreversible ETCS data loss. However, considering that the mean ETCS delay was significantly shorter than the maximum acceptable 500 ms, it should be possible to retransmit the lost packets without exceeding the delay requirements. The retransmission mechanisms for preventing ETCS data loss and ensuring ETCS data integrity are investigated in Section 5.4 on page 132.

It should be also noted, that the packet loss results did not reach stable values. The reason for that was the random distribution of trains at the station. In some

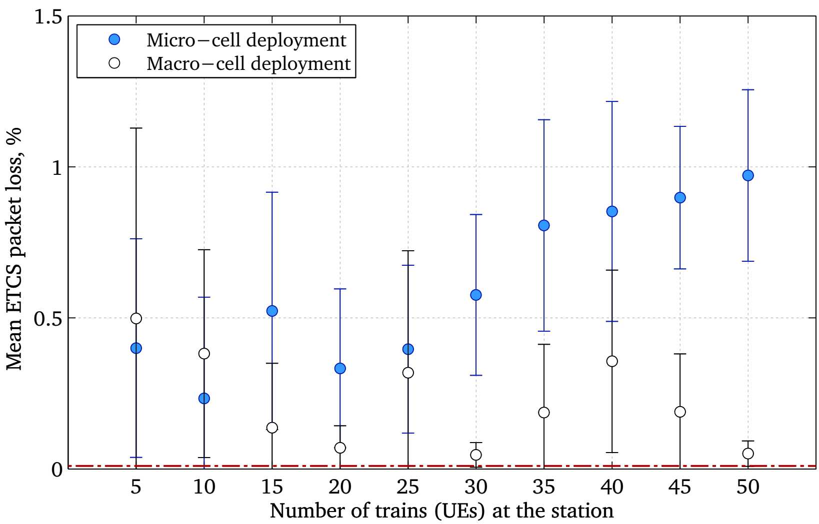


Figure 5.7: ETCS packet loss in the two alternative deployments. Red dashed line indicates the maximum data loss acceptable by ETCS. Error bars indicate 95% confidence intervals.

simulation cases, trains were located closer to the base station, thus, the transmission performance was better. In other cases, train locations were more unfavourable and the transmission performance was worse. Therefore, the packet loss issue requires further work and additional simulations.

Voice communication

Another application whose transmission performance was analysed in the simulations was the railway voice communication (telephony). Similarly as in case of ETCS, the application performance was measured in terms of delay and loss.

Figure 5.8 shows the mean voice packet delay in relation to the number of trains. The measured delay was the “mouth-to-ear” delay. Hence, it includes both the network transfer delay and the encoding/decoding delay (see Section 4.1.2 on page 89 for more details). Due to that, the results are significantly higher than in case of ETCS, where only the transfer delay was considered.

In the macro-cell deployment, the mean voice delay was between 104 and 106 ms. In the micro-cell deployment, it was between 106 and 109 ms. In both scenarios, the more trains were at the station, the higher was the delay. However, the delay increase was small and it would have no impact on the communication quality, e.g. in terms of the perceived sound quality. Moreover, in both scenarios, the mean delay was below the 150 ms limit [83] in all of the investigated cases.

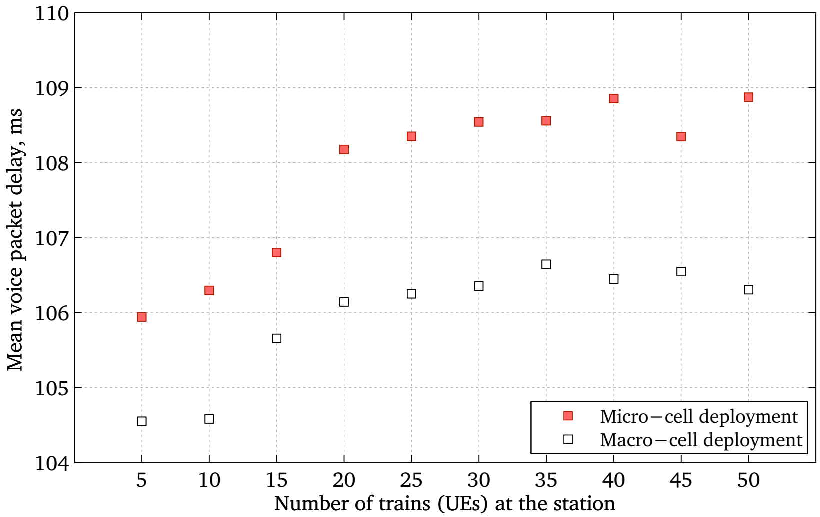


Figure 5.8: Voice “mouth-to-ear” delay (including transfer and coding/decoding delay) in the two alternative deployments

All in all, both deployments offered very similar performance, so the difference between them would not be noticeable for the voice communication users.

Figure 5.9 shows the mean voice packet loss in relation to the number of trains. In the macro-cell deployment, the mean packet loss was in the range between 0.5% and 1.4%. In the micro-cell deployment the loss was between 0.1% and 1.9%. In the micro-cell scenario, the more trains were at the station, the higher was the voice packet loss. This was due to the inter-cell interference, which grew with the number of trains at the station. However, similarly as in the case of ETCS packet loss, the results did not reach stable values.

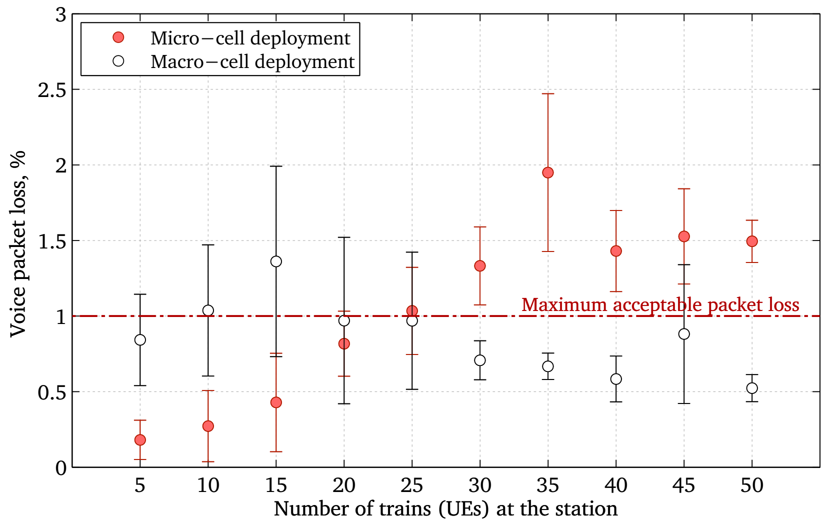


Figure 5.9: Voice packet loss in the two alternative deployments. Error bars indicate 95% confidence intervals.

Video surveillance

The last application whose performance was investigated in the simulations was the video surveillance. In contrast to the two previous applications, this one is non-critical from the railway point of view. Due to that, this application was delivered over the default EPS bearer, which does not guarantee any dedicated resources or transmission performance quality (e.g. in terms of delay).

Figure 5.10 shows the mean video transfer delay in the two deployments. According to the requirement [94, Sec. 5.5, Table 1], the maximum acceptable mean delay is 150 ms. The macro-cell based deployment offered performance that fulfils this requirement only in the first two cases: with 5 and 10 trains at the station.

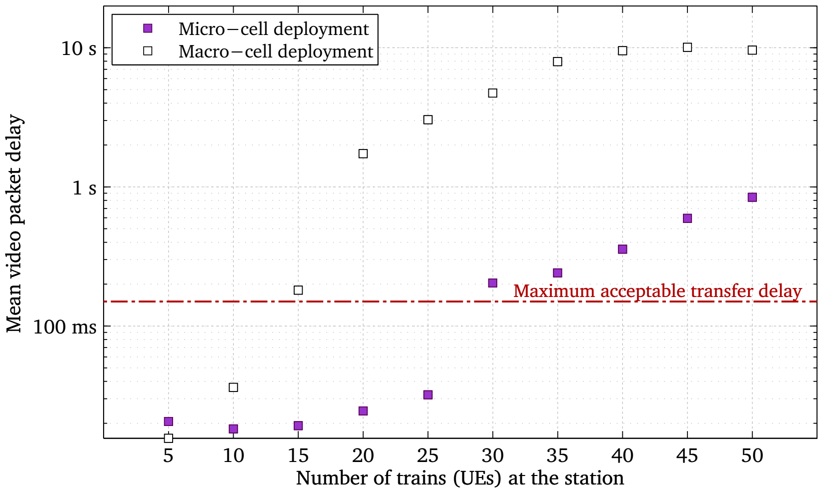


Figure 5.10: Video packet transfer delay in the two alternative deployments. Note the logarithmic scale on the vertical axis.

In the cases with more trains, the delay began to increase rapidly. Already in the case with 20 trains, the delay exceeded 1 s. This was due to the insufficient radio capacity of the macro-cell based deployment.

In the micro-cell deployment the video delay was significantly lower. Up to the case with 25 trains, the mean delay did not exceed the 150 ms limit. This improvement was due to the higher throughput available in this deployment.

Figure 5.11 shows the mean video packet loss in relation to the number of trains. Both deployment offered unsatisfactory performance in terms of packet loss. Almost in all of the cases, the 1% limit [94, Sec. 5.5, Table 1] was exceeded.

5.3.4 Discussion of the results

The simulations that were presented in this section compared the two alternative radio deployments. The first one, which was based on the macro-cells, was a typical deployment that is used in the current railway mobile networks. The second deployment, which was based on the micro-cells, was the modelling the new radio layer proposed in the heterogeneous network architecture.

The biggest difference between the two deployments was in the available radio capacity. The macro-cell deployment offered a maximum throughput of approximately 13 Mbit/s. It was sufficient to provide ETCS signalling and railway voice communication, due to the low bandwidth requirements of these applications and due to the effective QoS mechanism. Thus, the micro-cell deployment is able

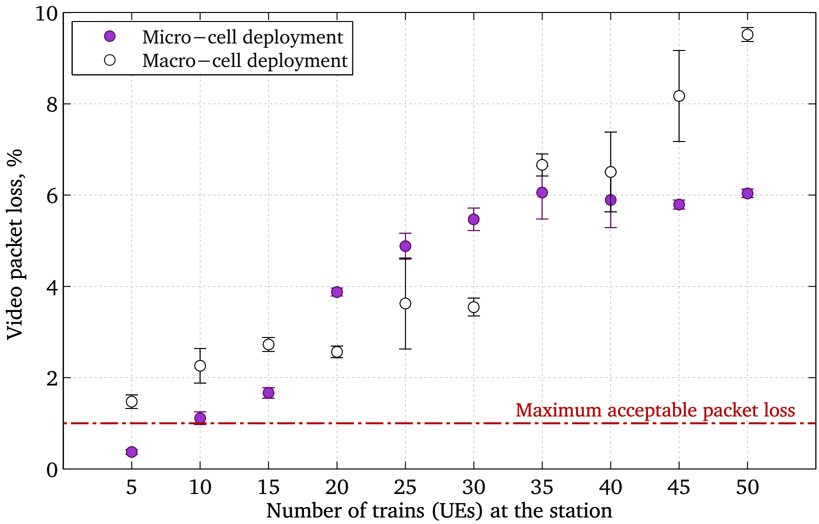


Figure 5.11: Video packet loss in the two alternative deployments. Error bars indicate 95% confidence intervals.

to provide these two critical applications for at least 50 trains, which is more than expected at Copenhagen Central Station.

However, the capacity of the macro-cell deployment may not be sufficient to provide bandwidth-demanding applications, as illustrated with the example of the video surveillance. Since each train generated a 1 Mbit/s uplink video stream, the radio capacity was exceeded already in the case with 15 trains at the station. The modelled video surveillance was only an example of a bandwidth-demanding application, but it demonstrated that the macro-cell capacity is limiting the choice of applications that railways may use.

Since the micro-cell deployment consisted of many more base stations, it offered a significantly higher radio capacity. This did not have a major impact on the transmission performance experienced by ETCS and the voice communication. Thus, neither of these applications should be affected by the small difference between the two deployments (assuming stationary nodes).

Nevertheless, the additional capacity of the micro-cells was very beneficial for the video surveillance. Up to the case with 25 trains, the transmission performance requirements were fulfilled. Thus, the micro-cell deployment, in comparison to the macro-cell, is able to offer the video application to over twice as many trains.

In both deployments, but especially in the micro-cell based, there was an issue of a high packet loss. This affected all of the applications: both critical and non-critical. Such a high packet loss is unacceptable especially for ETCS signalling.

Therefore, the following section investigates the packet loss issue and the prevention mechanisms.

5.4 Ensuring ETCS data integrity in LTE micro deployment

One of the most important ETCS transmission requirements is the integrity of the data sent between the OBU and the RBC. ETCS requires the data loss probability to be below 10^{-4} [19, p. 18]. In the simulation results presented in Section 5.3.3, the observed ETCS packet loss (and the consequent data loss) exceeded this limit significantly. Therefore, the ETCS data integrity in LTE must be investigated in detail. This was the purpose of the research work that is presented in this section. More specifically, the goals were to:

- Model a railway LTE network in the worst-case scenario in terms of: traffic load, base station density, high UE concentration, and their unfavourable positions in respect to the base stations.
- Validate performance of LTE and ETCS retransmission mechanisms in terms of minimizing the ETCS data loss probability.
- Compare the data loss results with ETCS data integrity requirements in order to verify if LTE can ensure reliable ETCS communication despite an unreliable physical transmission.

The research work presented in this section has been previously published in a paper [Sniady2015b].

5.4.1 Data integrity protection in LTE

In an LTE network the highest risk of data loss is on the radio link. Wireless transmission, due to its nature, can be disrupted by electromagnetic noise, interference or signal power variations (e.g. due to fading) [76, p. 53]. These various physical effects may introduce errors in the transmitted radio signal that make it unrecognisable at the receiver. As a result, the transmitted data is lost.

Link adaptation

The probability of a transmission error on the radio link depends on the Signal-to-Interference-and-Noise Ratio (SINR) and on the Modulation and Coding Scheme (MCS) chosen by the network [38, p. 218]. A low SINR value means that the interference and the noise are relatively strong compared to the LTE signal. Due to that, in low SINR conditions, the probability of a radio error is high. In response to poor radio conditions (low SINR), LTE selects a more robust MCS. This reduces the error probability, but it also reduces the radio throughout. Thus, the network must constantly adjust the MCS in order to provide the highest possible throughput while keeping the error probability at a stable and acceptable level [38, p. 217]. This balancing between the throughput and the error probability is the basic purpose of the link adaptation mechanism.

The error probability is defined as the BLock Error Rate (BLER), i.e. the percentage of the erroneous radio blocks. In LTE, the typical BLER target for user data is 10% [39, p. 125]. This means that up to 10% of the received radio blocks may be erroneous. BLER target cannot be 0%, because there is always an inevitable probability that a wireless transmission is disrupted.

Moreover, then non-zero BLER is beneficial for the radio throughput. In order to maximize the throughput and fully utilize the radio capacity, a small error rate on the radio link is desired [25, p. 230]. This comes from the fact that if the higher error target is set, the less conservative MCS are selected by the network. On one hand, this increases the error probability, so a higher portion of the radio blocks is lost. But on the other hand, the throughout is higher, so the successfully received blocks deliver more data. According to Sauter [25, p. 230], the 10% BLER target maximizes the radio throughput.

Since the errors on the radio link are expected and even desired, LTE offers two retransmission mechanisms for recovering the data lost on the radio link. One mechanism is on Medium Access Control (MAC) layer and the other on Radio Link Control (RLC) layer. Moreover, in the particular example of ETCS, additional end-to-end retransmission mechanism is provided at the application layer (Euroradio). These three mechanisms for data loss prevention, i.e. data integrity protection, are illustrated in Figure 5.12 and described in the following sections.

MAC layer retransmissions

In LTE, the principle data protection mechanism is the *Hybrid Automatic Retransmission Request (HARQ)*, which is implemented at radio MAC layer [38, pp. 108–109]. HARQ protects all traffic transmitted over the radio link, regardless of which EPS bearer they are sent over.

HARQ operates in a stop-and-wait fashion. Thus, after each transmitted radio transport block, the sending process stops the transmission and waits for a single-bit ACK/NACK feedback from the receiving process. If an ACK is received, a new portion of data is sent. On the other hand, if a NACK is received, the previous data is retransmitted. In the uplink HARQ, there is a synchronization between the transmitted data, the ACK/NACK feedback and the following retransmission attempts [38, p. 241]. Thus, each retransmission attempt adds a minimum delay

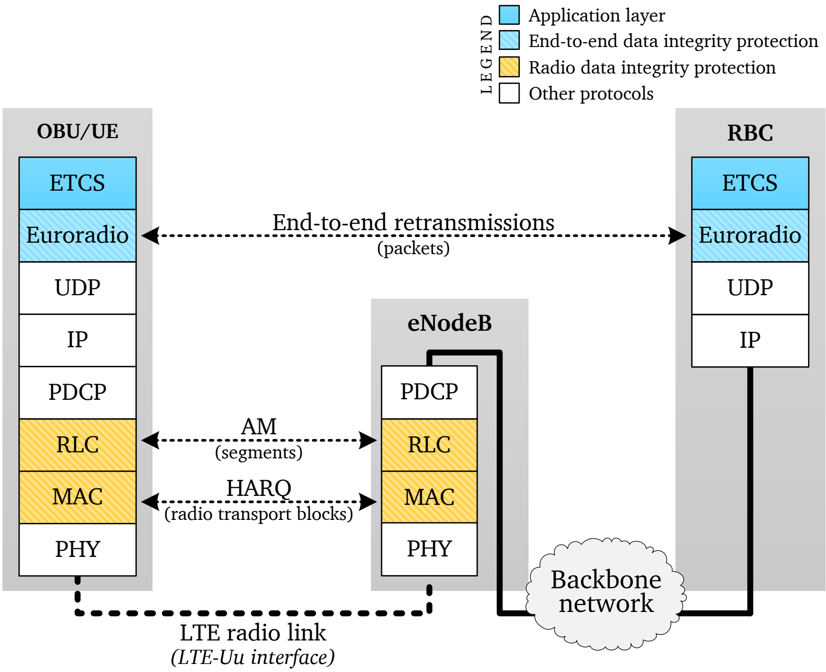


Figure 5.12: Retransmission mechanisms that are used for protecting ETCS data integrity

of 7 ms [39, p. 118]. In order to allow LTE nodes to transmit continuously, eight HARQ processes are used in parallel. While one process is sending data, the other seven are waiting for their respective ACK/NACK feedbacks.

LTE HARQ supports *incremental redundancy*. This means that, at each retransmission attempt, the radio block can be sent with more redundant bits in order to increase the probability of a successful decoding (i.e. reception). Moreover, the receiver combines the signals received in each attempt. This method is called *soft combining* [39, p. 118]. Thanks to these two, the decoding probability increases with each retransmission attempt.

The maximum number of HARQ retransmission attempts is limited. Moreover, since the ACK/NACK feedback in is only one bit long, there is a relatively high probability that it is received erroneously itself [76, p. 54]. Therefore, after HARQ, there is still a residual data loss left that must be solved by the higher layers.

RLC layer retransmissions

RLC layer, which is just above the MAC layer, operates in one of three modes, which are configurable independently for each EPS bearer [38, pp. 98–107]:

- *Transparent Mode (TM)*: The RLC layer only forwards data packets between the upper and the lower protocol layers.
- *Unacknowledged Mode (UM)*: At the sending node, the data packets are segmented and passed to the MAC layer [76, p. 53]. Segment size is determined by the current radio transport block size. At the receiving node, segments arriving from the MAC layer are reassembled into the original packet. If any segment is missing, then the packet cannot be reassembled and the data is lost. Thus, UM depends entirely on HARQ to recover the segments lost during a radio transmission.
- *Acknowledged Mode (AM)*: Data packets are segmented and reassembled in the same way as in UM operation. However, AM offers an additional retransmission mechanism. Thus, in case that the MAC layer fails to deliver a segment, the missing segment may be retransmitted.

Since, the RLC mode is chosen for each bearer, sensitive data can be sent in AM, while other data can be sent in UM. For instance, ETCS signalling may be delivered in AM in order to provide lower data loss probability.

ETCS layer retransmissions

Besides the two LTE mechanisms, ETCS data is protected by the end-to-end retransmission mechanism at the Euroradio layer (see Section 3.2 on page 35). Whenever the OBU or the RBC sends an ETCS message, they expect a 5-byte ACK reply, which confirms that the message arrived successfully to the receiver. If the ACK does not arrive within a configurable period, the ETCS message is retransmitted. The details on operation this mechanism were presented in Section 3.6.1 on page 46. The purpose of this mechanism is to address:

- The data losses occurring in other part of the network than the radio link. These losses can be caused by buffer overflows, software errors, etc.
- The cases when the maximum number of the MAC and the RLC retransmission attempts is reached, but the data is not delivered successfully over the radio link. Without the end-to-end retransmissions, the data would be lost.

In the ETCS model used in the following simulations, Euroradio retransmission functionality was included within the ETCS layer. Thus, the end-to-end mechanism is referred to as “ETCS retransmissions”.

5.4.2 Simulation scenarios

In order to validate performance of these three retransmission mechanisms, a set of simulation scenarios was prepared. The goal of these simulations was to:

- Model a railway LTE network under the most challenging conditions in terms of the base station density, traffic load and UE distribution.
- Measure ETCS data loss probability under these worst case conditions.
- Model various configurations of the three retransmission mechanisms and measure their impact on ETCS data loss.
- Compare the results with ETCS data integrity requirement.

The simulation setup was based on the micro-cell deployment, which was presented in Figure 5.4b on page 122 and described in Section 5.3.1 on page 121. In this deployment, the modelled LTE network at Copenhagen Central Station consisted of 10 eNodeBs placed every 100 m along the station tracks, i.e. the cell radius was 50 m. The micro-cell deployment is more challenging than the macro-cell deployment, because when a train station is covered with many small cells, then there is a risk that many trains (UEs) will happen to stop near a cell edge. As explained in the next section, this is very unfavourable for the radio transmission, especially in the LTE network.

Train distribution

As in the previous setups, every train was modelled as an UE. In the simulations, 40 trains were present at the station. This is approximately the maximum train traffic expected at Copenhagen Central Station in 2030 (see Eq. 3.10 on page 71).

In order to model the most challenging radio conditions, all trains were placed at the edges between the neighbouring micro-cells. This setup is illustrated in Figure 5.13. Such a train (UE) distribution is the least favourable for radio transmission. This is because, relatively high transmission power has to be used to compensate for the signal path loss. Moreover, trains attached to different eNodeBs are located near each other. All of the LTE cells within the same LTE network operate at the same frequency [38, p. 287]. Thus, the UEs from the neighbouring cells introduce a interference in each others transmission. This effect is the strongest (i.e. the worst) near the cell edge.

In this simulation model, all trains were considered as stationary nodes. Thanks to this simplification, it was possible to repeat simulation runs maintaining the predefined unfavourable train distribution. Thus, any differences observed in the collected results were due to the changes in the configuration, not due to the randomness of the train distribution. Besides, in a station area most of the trains

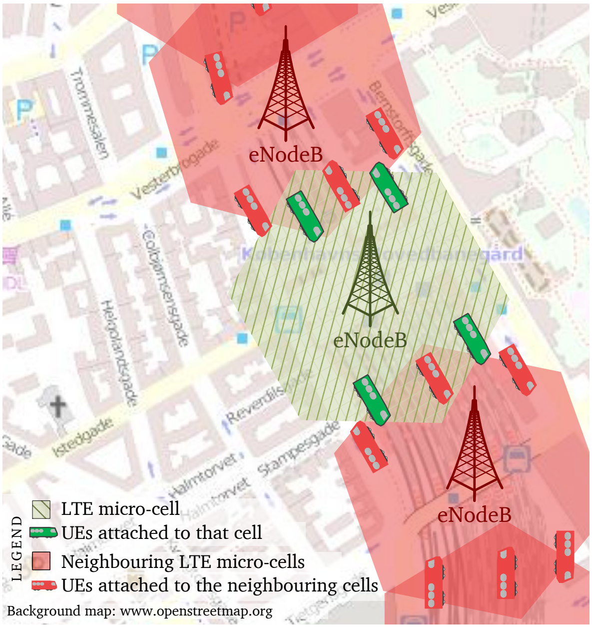


Figure 5.13: Train distribution considered in the simulations. Concentration of trains (UEs) at the cell edges is the least favourable from the point of view of radio transmission.

are stopped or drive at a low speed. Thus, the speed is not a major factor affecting the transmission performance.

Application mix

The application mix used in the simulations consisted of ETCS signalling, the voice communication, the voice announcements, and the video surveillance (uplink). All the applications were configured as described in Section 5.3.2 on page 123. The only exception was an abandonment of the tele-maintenance application, which generated a very rare, but very high traffic load. Due to this exceptionally bursty application, in the previous simulations, it was difficult to reach the stable mean values of the analysed statistics.

Instead of the tele-maintenance, the video surveillance was used as a source of background traffic. Since this video application was streaming continuously, it was better for modelling traffic load on the network. Also, railways expressed some interest in introducing real-time video transmission within station areas [11, p. 6]. Thus, it is a possible application to be used by railways in the future.

Traffic load scenarios

Traffic load is one of the most important factors expected to affect the packet loss and the consequent ETCS data loss. Therefore, three simulation scenarios were considered with different traffic load on the network. This difference in the load was controlled by the number of video streams sent by the trains:

- **Scenario 1 with light traffic load**: The video surveillance was disabled in this Scenario. Thus, there was no video traffic in the network. The mean utilization of the PUSCH was only 1.46%.
- **Scenario 2 with medium traffic load**: Four trains were transmitting uplink video streams to a station security centre, which was modelled as an application server. Each stream had a 1000 kbit/s bitrate. In the four cells where the trains with video surveillance were located, the mean utilization of the PUSCH was 65.79%. The mean utilization in all 10 cells was 35.24%.
- **Scenario 3 with traffic overload**: Forty video streams were transmitted in the network, i.e. one stream from each of the 40 trains. The mean PUSCH utilization was 98.87%.

Retransmission configuration cases

The three Scenarios were evaluated under different configurations of the retransmission mechanisms. As shown in Table 5.3, five configuration cases were considered:

- **Case A**: Only the MAC layer retransmissions (HARQ) were enabled. A maximum of three retransmission attempts was allowed. The RLC layer operated in the UM, so it did not provide any data protection mechanism.
- **Case B**: RLC operated in the AM, thus, RLC retransmissions were used on top of MAC retransmissions. A maximum of three retransmission attempts was allowed on each layer.
- **Case C1–C3**: ETCS end-to-end retransmissions were enabled besides the two mechanisms at the lower layers. Therefore, all three data protection mechanism were used in this case. A maximum of three retransmission attempts was allowed on the MAC and the RLC layers. On the ETCS layer, either one (case C1), two (case C2) or three (case C3) attempts were allowed.

Other parameters used in the simulations are summarized in Table 5.4.

Table 5.3: Retransmission configuration cases. The maximum number of retransmission attempts at each layer is presented. A dash means that the retransmission mechanism was disabled in this configuration.

Case	**MAC layer**	**RLC layer**	**ETCS layer**
A	3	—	—
B	3	3	—
C1	3	3	1
C2			2
C3			3

Table 5.4: Simulation parameters and configuration

Parameter	**Value**
Carrier frequency[1]	5.9 GHz
Bandwidth	5 MHz
eNodeB antenna height[2]	10 m
eNodeB antenna gain[3]	15 dBi
UE antenna height[4]	4 m
UE antenna gain[3]	1 dBi
Path loss model[5]	ITU-R M2135 Urban Micro (UMi)
Multipath channel model[6]	ITU Pedestrian A
Link adaptation BLER target[7]	0.01%

[1] ITS radio band [92] was used as an exemplary high-frequency band, as explained in Section 5.2 on page 118.

[2] Assuming that the eNodeB antennas are attached to the station's ceiling.

[3] Chosen within the typical range as given in [39, p. 223].

[4] Assuming that the UE antenna is placed on a train roof [22, p. 41].

[5] UMi path loss model was chosen, because it is applicable to densely urbanized areas, such as the one around Copenhagen Central Station. Moreover, it supports the 5.9 GHz carrier frequency [93].

[6] Since the trains had fixed positions, low-speed channel model was used.

[7] Link adaptation was configured with a significantly lower BLER target than the typically used 10% [39, p. 125]. Due to that, the more conservative and error-prone MCS were chosen for the radio transmission.

5.4.3 Simulation results

The three scenarios and the five configurations gave a total of 15 simulation cases to consider. Each of them was executed at least 70 times with different seed numbers in order to reach stable average results. Single simulation run lasted 15 minutes. Figure 5.14 shows the ETCS data loss probability estimated based on the simulation results. The following sections discuss the observed ETCS data loss and its impact on ETCS operation.

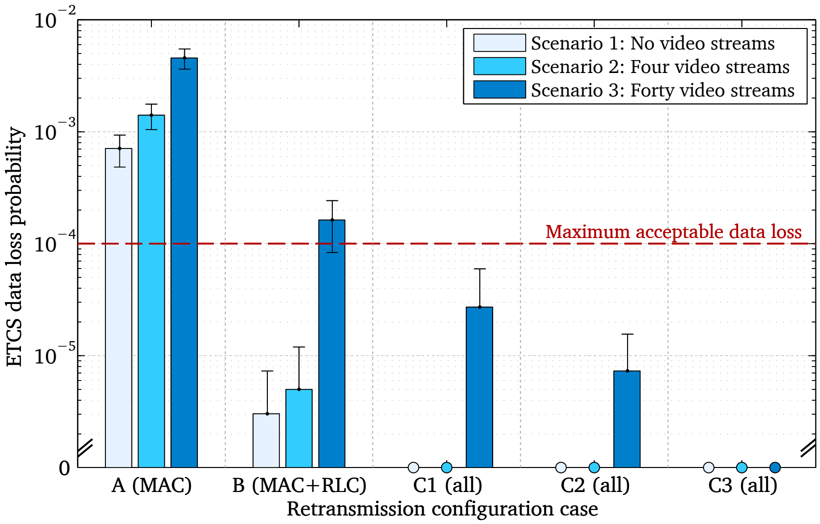


Figure 5.14: Impact of the retransmission mechanisms on the ETCS data loss probability. Error bars indicate 95% confidence intervals. A dot on the horizontal axis means that data loss was not observed in this particular configuration.

Although the data loss probability observed in the simulations was high, it is in accordance with the results published in various lab and field tests of LTE networks. For example, in measurements made by Anehill et al. [82, p. 9], the packet loss at the LTE cell edge was 4×10^{-3}. In other field trials made by Chen et al. [95], the packet loss was in a range between 3×10^{-4} and 2×10^{-3}. Thus, the results observed in the simulations are realistic, especially considering the worst-case assumptions made in the setup.

Scenario 1 results

The first data series in Figure 5.14 shows the results from Scenario 1. Since the video surveillance was disabled in this scenario, the traffic load on the network was the smallest. Only the low-bandwidth applications were provided: ETCS signalling, the voice communication and the voice announcements.

In *case A*, when only the default HARQ mechanism was used, the ETCS data loss probability was 7.08×10^{-4}. Thus, it was above the maximum acceptable 1×10^{-4}. As expected, the data loss probability in case A was the highest among the considered configurations, because in this case only one retransmission mechanism was enabled. If the HARQ failed to deliver a radio block, then the data that was carried in that block was irreversibly lost.

In *case B*, the data loss probability was 3.02×10^{-6}, which was 99.6% lower than the probability in case A. This significant reduction was achieved by using the RLC layer in the AM, which retransmitted those segments that were not delivered by the MAC layer. Although some data loss was observed, it was below 1×10^{-4}, so the ETCS data integrity requirement was fulfilled.

In *case C1*, one end-to-end retransmission attempt was allowed. This was sufficient to address the remaining data loss. As a result, all of the sent ETCS messages were successfully delivered and no data loss was observed.

Scenario 2 results

In Scenario 2, the traffic load on the network was slightly increased by introducing the four uplink video streams. As visible in Figure 5.14, the results observed in this scenario were higher than in Scenario 1, but they followed the same trend in response to the configuration changes.

In *case A*, the data loss probability was 1.40×10^{-3}. Then, in *case B*, the probability was reduced by 99.6%, down to 5.00×10^{-6}. In *case C1*, no ETCS data loss was observed.

Scenario 3 results

In Scenario 3, all of the 40 trains were transmitting video in uplink. Due to that, the traffic load exceeded the uplink radio capacity of the modelled LTE network. In a properly dimensioned network such a heavy traffic overload should not happen. However, the purpose of this scenario was to model the worst-case conditions, which are not expected in the everyday network operation.

In *case A*, the data loss probability was 4.56×10^{-3}. It was considerably higher than it is acceptable by ETCS. Moreover, the loss probability was higher than observed in the previous Scenarios 1 and 2.

In *case B*, the loss probability was lowered down to 1.63×10^{-4} (reduction of 96.4%). In opposite to the previous scenarios, under heavy traffic load, the RLC retransmissions were not effective enough to fulfil the ETCS requirement. The data loss probability was still above the 1×10^{-4} limit.

In *cases C1-C3*, the end-to-end retransmission mechanism was used. With each additional retransmission attempt, the data loss probability was gradually falling. In case C1, the probability was 2.71×10^{-5}, then in case C2 it was 7.31×10^{-5}. Finally, in case C3, no data loss was observed.

5.4.4 Discussion of the results

Two important features of the simulation setup must be emphasized. Firstly, all cells operated in the same frequency band (frequency reuse factor equal to one) as typically in LTE [38, p. 287]. Secondly, all trains (UEs) were concentrated at the cell edges, as was shown in Figure 5.13 on page 137. As a result, there was a strong interference between radio transmissions in the neighbouring cells. Consequently, the SINR was low and the BLER was high. Hence, the probability of a transmission failure on the radio link was high. The transmission failures on the radio link and the consequent radio block loss were the causes of the ETCS data loss. This is why the additional retransmissions on the radio link offered by the RLC AM (case B) were effective in lowering the loss probability.

Even a small loss on the radio link may cause much higher loss at the application layer. This is caused by the segmentation mechanism at the RLC layer. A single ETCS packet may be split into multiple RLC segments, which are then transmitted in separate radio blocks. If any of these blocks is lost, then the segment is lost and the packet cannot be reassembled [38, p. 101]. Thus, even though remaining segments are received correctly, they must be discarded and the entire packet with ETCS data is lost. Moreover, it may also happen that a single radio block carries segments belonging to multiple packets. A loss of such a block causes a loss of all of those packets.

Comparing the results from the three scenarios, for example in case A, it is visible that the higher was the traffic load, the higher was the data loss probability. This is because, when the traffic load was growing, the interference and BLER were growing as well.

Although ETCS traffic was carried over a high-priority EPS bearer, LTE QoS mechanism could not prevent the data loss. Following the bearer configuration, eNodeB packet schedulers prioritized ETCS packets over other applications' packets. However, eNodeBs manage their radio resources independently. Therefore, even if an eNodeB assigned specific radio resources for ETCS transmission, it could not prevent the neighbouring eNodeBs from using the same resources. Considering the high traffic load and the chosen train distribution, there was a high probability that the resources carrying ETCS data were interfered by the neighbouring cells.

In the simulations, three different mechanisms were applied for the purpose of reducing the ETCS data loss. The default mechanism, i.e. the HARQ at the MAC layer, did not manage to provide sufficiently low data loss probability to comply with the ETCS requirement. Therefore, HARQ had to be supported by an additional mechanism at the RLC layer.

Whereas HARQ retransmitted all traffic, the RLC retransmissions were used only for ETCS. Moreover, the RLC mechanism is aware of the packet segmentation, it can request retransmission of the specific segments that were lost on the radio. Also, RLC relies on status messages for retransmission requests. These are significantly less error-prone that a single bit ACK/NACK feedback of the HARQ [76, p. 54]. Due to these, the configuration in case B was much more successful in minimizing the data loss. Only in the case with heavy traffic load, RLC AM did not manage to fulfil the ETCS data integrity requirement.

Under overload conditions, as in Scenario 3, the end-to-end retransmissions were necessary to lower the data loss probability down to the acceptable level. However, regardless of the traffic load and the loss on the radio link, the end-to-end mechanism is necessary. This is because, data loss can occur in other parts of the network. MAC and RLC mechanisms can recover only the data that is lost on the radio link. Thus, the retransmission mechanism at the application layer is required to prevent ETCS loss occurring in other parts of the network.

The retransmission mechanisms countermeasure the loss on the radio link and, therefore, lower the ETCS data loss probability. However, ETCS data integrity could be also improved by eliminating or minimizing the cause of the data loss, i.e. the inter-cell interference. Multiple LTE features could be used to lower the interference, for instance [38, pp. 287–290]:

- *Inter-Cell Interference Coordination (ICIC)* is a mechanism coordinating radio resource usage between the neighbouring cells. ICIC is build around the concept of partial frequency reuse. Thanks to it, eNodeBs use different frequencies near the cell edge. This considerably reduces the interference for the UEs that are in this interference-sensitive location.

- *Coordinated Multi-Point (CoMP)* is a mechanism that combines transmissions from multiple eNodeBs. Thanks to CoMP, a signal from the neighbouring eNodeB improves the signal received by an UE, instead of interfering with it.

- *Carrier Aggregation (CA)* is a mechanism introduced in LTE-Advanced (LTE-A), CA aggregates multiple LTE frequency channels in order to increase the available radio bandwidth. The main purpose of this mechanism is to increase throughput. However, thanks to CA, the traffic load is also distributed over a wider frequency range. Assuming that the traffic load is unchanged, the radio utilization and the interference is lower in a network with CA than in a single-carrier network.

Data loss impact on ETCS

In Section 3.1 on page 34, it was explained how a reliable communication between the train On-board Unit (OBU) and the Radio Block Centre (RBC) is the basis of the ETCS system operation.

One of ETCS procedures that are the most vulnerable to communication disruptions is Movement Authority (MA) update. MA is a command which defines, among other things, the distance that a train is allowed to drive. The last point where the must stop is called End of Movement Authority (EoMA). The train cannot pass the EoMA unless a new MA is delivered with a new EoMA. As the train is running and approaching its current EoMA, the OBU should receive the new MA from the RBC. If the MA is not delivered, for example due to a communication failure, then the train is forced to slow down and eventually to stop [53, pp. 23–31]. Due to the specifics of railways, such a stop may cause a knock-on delay that propagates through the network and causes operational disruptions for other trains [54]. Hence, ETCS data loss during MA update may be very disruptive for railway operation.

In order to ensure uninterrupted operation, the MA should be sent to the OBU in some time advance before the train reaches the EoMA. This adds a time margin, which can be used, for example, to accommodate possible communication problems and the consequent data retransmissions. On the one hand, the bigger is this margin, the higher is the probability that ETCS messages are successful deliver without any impact on the train operation. On the other hand, the later the MA is sent, the shorter is the time the train "occupies" given track section. Thus, track capacity is improved.

ETCS-based railway signalling systems are build under an assumption that the MA update procedure takes 4 to 5 s, on average, and 12 s, maximum. Thus, if the OBU receives the MA within 12 s, the unnecessary braking is avoided [53, p. 32]. Internal operations of ETCS elements may take approximately 2.5 s. The remaining time can be used by the communication network. Thus, MA transfer must be below 1.5 s, on average and 9.5 s, maximum.

From the point of view of ETCS system, MA transfer delay (denoted here as t_{MA}) consists of two elements. Firstly, there is the delay due to each unsuccessful transmission attempt. This delay depends on the time-out of the end-to-end retransmission mechanism ($t_{timeout}$) and the number of attempts (n). Secondly, there is the delay due to the successful transmission through the network ($t_{network}$). All in all, t_{MA} can be estimated as:

$$t_{MA} = n \cdot t_{timeout} + t_{network} \tag{5.1}$$

In the simulations, the retransmission time-out was set to 500 ms. Maximum three attempts were necessary to deliver an ETCS message successfully. The maximum delay in the network was 1.7 s (measured in case A, with no retransmissions).

Therefore, the worst-case t_{MA}, as estimated using Equation 5.1, should be approximately 3.2 s. This is considerably lower than the 9.5 s, which is used as the worst-case assumption in the signalling systems. This means, that even if multiple end-to-end retransmissions are necessary to successfully deliver the MA, the delay due to these retransmissions should not cause interruptions in the train operation.

Data loss due to external factors

Since neighbouring LTE cells operate at the same frequencies, their performance is closely correlated by the mutual interference [25, p. 260]. Transmissions in one cell impact the radio error probability in the neighbouring cells, as was demonstrated in the simulations. The research work presented in this section considered ETCS data loss due to these internal inter-cell interference in LTE.

However, apart from the interference caused by the LTE network itself, external factors may also be the cause of ETCS data loss. For example, these factors could be a strong electromagnetic interference, a software failure or a hardware failure. Retransmission mechanisms often cannot address disruption caused by these external factors. For instance, a strong interference from an external source may corrupt all radio communication. If this interference lasts long time, all retransmission attempts may be fruitless. Therefore, other solutions should be applied to address such external factors. One of them could be the redundant heterogeneous radio architecture, as presented in Section 5.2 on page 117. If one of the radio layers (macro and micro) is interfered, then the communication may be handed over to the other layer. Thus, a reliable ETCS communication requires both: robust mechanisms protecting the transmission and a resilient network architecture.

5.5 Chapter conclusions

Since railways are increasingly dependent on the communication-based applications (e.g. ETCS), the high availability of these applications becomes a fundamental requirement. In turn, this availability depends on the reliability of the underlying communication network, which must be resilient against failures and must offer sufficient transmission capacity.

The traditional railway radio architectures often use redundant solutions in order to increase the network resilience. On the other hand, being based on large macro-cells, these traditional architectures offer a relatively low radio transmission capacity. This may be sufficient for the critical-applications, such as ETCS and voice communication, but the non-critical applications may require alternative higher-capacity solutions. Therefore, in this chapter, a novel heterogeneous radio access architecture for railways was proposed. Its purpose is to evolve the traditional architecture in order to offer additional benefits besides the resilience. The new heterogeneous architecture, which is based on a combination of macro and micro radio cells, should: ensure high network availability, increase radio capacity, and optimize radio cell deployment for different types of railway users. Thanks to the second micro-cell based radio layer, railways would have an opportunity to introduce new bandwidth-demanding applications, such as video surveillance.

The new heterogeneous architecture can be considered as a migration strategy during the likely introduction of a GSM-R successor [33, p. 7]. Moreover, the proposed architecture allows mainline railways to use high-frequency radio bands which were usually not considered due to their poor propagation properties. Micro-cell based architecture opens the possibility to use these bands and the wireless technologies that operate in them, such as IEEE 802.11 (Wi-Fi). This particularly technology is already requested by some railway stakeholders [33, p. 47]. Besides, an introduction of Wi-Fi in mainline railways could be an important step in the possible future convergence between ETCS and CBTC-based railways.

Despite the numerous benefits of the proposed heterogeneous architecture, it may also bring some challenges due to the dense deployment at the micro level. The simulation results indicate that in a dense LTE network, under a heavy traffic load, ETCS data loss may exceed the acceptable limits. Therefore, the second part of the chapter investigated data protection mechanisms on the radio and on the end-to-end layers. The outcomes of this investigation demonstrate that with a set of properly configured retransmission mechanisms, ETCS data integrity can be ensured even in the worst-case scenario.

CHAPTER 6

Conclusions and Outlook

GSM-R has brought railway telecommunications into the digital era. In comparison to the previous analogue systems, it has been a revolution in terms of technology as well as international interoperability. Since GSM-R replaces a wide range of incompatible country-specific standards, it is one of the fundamental elements enabling uninterrupted cross-border operation. Moreover, GSM-R is the basis of the two most important railway applications: ETCS signalling and the voice communication (including REC).

On the other hand, since GSM-R was designed, a few important changes have occurred in the fields of railways and telecommunications. GSM-R capacity has turned out to be insufficient for the current and the future train traffic, especially in the busy areas, such as train stations. Railways have also noticed the potential benefits that could be derived from innovative applications, which cannot be provided over GSM-R. From the point of view of telecommunications, GSM-R is an outdated technology lacking in terms of efficiency, capability and capacity. Besides, the industry support for GSM-R is expected to decrease.

GSM-R is turning into a bottleneck for railway operation and an obstacle for innovation. Due to its limited capabilities and capacity, GSM-R is not able to fulfil the growing communication demands of railways. Therefore, alternative technologies must be considered to replace GSM-R in the future.

LTE as a railway communication technology

Railways need a mobile technology that is more efficient, offers higher capacity and has much lower obsolescence risk than GSM-R. It is generally agreed that railways, which are a relatively small industry, should base their networks on one of the broadly used telecommunication standards.

In this thesis, LTE was considered as a possible candidate for the future GSM-R replacement. LTE is significantly more efficient than GSM-R, offers large capacity, high throughput and low delay. Due to its advanced radio interface and packet-switched based transmission, it would allow European railways to take the best advantage of the limited radio spectrum they have available.

LTE as a railway mobile network must, first and foremost, deliver the critical applications, namely ETCS signalling and the voice communication. These two are essential for everyday railway operation. Therefore, the research work presented in this thesis investigated the transmission performance offered by LTE to these two critical applications. The results, which were collected in various simulation scenarios modelling typical railway conditions, demonstrate that LTE fulfils the transmission requirements set for both of the applications. This remained true regardless of the considered base station density, train speed, and traffic load.

Even under the worst-case conditions, LTE offers ETCS transfer delay approximately an order of magnitude lower than the maximum accepted by railways. ETCS data integrity requirements are also fulfilled despite the very unfavourable radio setting. Nevertheless, it must be noted that some LTE procedures, such as the random access and the uplink scheduling, are not optimal for ETCS traffic, which is infrequent and low-rate. Considering voice communication, VoLTE offers satisfactory setup time for railway one-to-one calls and RECs. Moreover, thanks to QoS mechanisms, LTE is able to differentiate between the call types and ensure REC prioritization. The sound quality in VoLTE is good, due to the low delay and loss of voice packets.

The simulation results show that LTE offers a significant capacity increase in comparison to GSM-R. A single radio cell can accommodate many more ETCS-equipped trains than it is expected even at the busiest Danish station until 2030. Additionally, taking into account the flexibility of LTE, the further capacity increase in LTE-A releases, and the low obsolescence risk, LTE should be a safe investment that could fulfil railway communication demand for many years ahead.

The high capacity and the effective QoS mechanism mean that LTE transmission resources can be safely shared between the critical and non-critical applications. Railways could introduce in their networks various new applications, such as tele-maintenance, ticketing, passenger information, and video surveillance. Neither ETCS signalling, nor the voice communication would be disrupted by the traffic generated by these non-critical applications.

It is worth noting that none of these businesses-supporting applications, individually, is essential or revolutionary enough to justify an introduction of a new communication technology. However, when these innovative applications are considered together, they may have a positive and profound impact on railway operation, its efficiency, safety, and attractiveness for the passengers. Thus, the communication-based applications may be an important advantage of railways in the competition with alternative means of transport.

Besides the transmission capabilities and the offered capacity, considerable assets of LTE are the standardized mechanisms for inter-working with other radio access networks, especially GSM-R. Due to the usual slow technology adoption in railways, GSM-R and the future railway communication networks will be coexisting for many years. In order to ensure uninterrupted train operation across the coverage areas of the old and the new networks, interoperability mechanisms between them are necessary. LTE offers standardized procedure for handovers to and from GSM-based networks. Moreover, LTE can flexibly share the radio band with GSM-R. These features should simplify the possible migration between the two technologies.

Heterogeneous networks as a likely future direction

The results presented in this thesis demonstrate that LTE is a viable alternative to GSM-R. However, considering the current trends in railways and telecommunications, it is likely that the future railway communication network will be based on multiple radio access technologies.

Therefore, the future solutions may be built on multi-level architectures, such as the macro/micro architecture proposed in this thesis. Apart from the mobile technologies (GSM, LTE), a supporting role may be played by Wi-Fi or satellite communication. These heterogeneous networks with complementary radio levels would allow railways to benefit from the particular advantages offered by different wireless technologies.

There are two conditions that must be fulfilled in a heterogeneous architecture. First of all, interoperability has to be ensured. The biggest advantage of GSM-R and ERTMS is that they are international standards, which greatly simplified cross-border operation. The future communication network must maintain this fundamental achievement. Secondly, railway applications must be offered indifferent to which radio technology is used at the moment. This is especially crucial in case of the critical applications. Decoupling of the applications from the underlying network is a necessary step for seamless application provisioning across a technologically heterogeneous architecture.

Non-technical factors determining the future

The future of railway mobile communication is undecided at this point. There are ongoing discussions and analyses on various strategical choices, such as: network ownership and control model, possible network sharing with public safety sector or other industries, roaming in commercial networks, and frequency band allocation [11, 33]. Although these issues are much more related to politics and business than to technology, the choices that will eventually be made will determine the future of railway mobile communication.

Therefore, whether LTE will become a railway communication technology depends on these strategic choices and the time they will be made. Nevertheless, due to its numerous advantages, LTE must be considered as a good candidate for the future railway communication network, regardless if it is going to be based on a single or multiple technologies.

Appendix A

Source code

This appendix includes detailed source codes of the application models used in the simulations. These models were prepared in AppTransactions Xpert (ATX) Whiteboard and they are written in Python programming language.

A.1 ETCS application model: Main process

The main process of the ETCS application model is responsible for establishing an ETCS session between an OBU and an RBC. Message exchange between the OBU and the RBC is shown in Figure 3.7 on page 45.

Initialization block

This script is executed when the process is initiated, i.e. before any ETCS messages are exchanged.

```
# Read the parameter values and verify whether they are valid
self.MAinterval = self.get_parameter ('MA_interval')
if self.MAinterval <= 0:
    self.sim_message ('The value for parameter "MA_interval" should be greater
        than 0.', 'Quitting the task.')
    self.quit ()

```

```
self.MAnumber = self.get_parameter ('MA_number')
if self.MAnumber <= 0:
    self.sim_message ('The value for parameter "MA_number" should be greater
        than 0.', 'Quitting the task.')
    self.quit ()

self.MaxRetransmissions = self.get_parameter ('MaxRetransmissions')
if self.MaxRetransmissions < 0:
    self.sim_message ('The value for parameter "MaxRetransmissions" should be
        greater or equal to 0.', 'Quitting the task.')
    self.quit ()

self.RetransmissionTimeout = self.get_parameter ('RetransmissionTimeout')
if self.RetransmissionTimeout <= 0:
    self.sim_message ('The value for parameter "RetransmissionTimeout" should
        be greater than 0.', 'Quitting the task.')
    self.quit ()

self.MAsize = self.get_parameter ('MA_size')
if self.MAnumber <= 0:
    self.sim_message ('The value for parameter "MA_size" should be greater than
         0.', 'Quitting the task.')
    self.quit ()
```

Function block

```
# Function responsible for the MA update procedure
def gen_ma_req (self, action, data):
    # Invoke a child process, which is responsible for sending a Movement
        Authority Request
    child_params = {'MAsize':self.MAsize}
    self.invoke_child_task(action, 'ETCSmovement.aed.m', False, child_params)
    self.sim_message('Time: ' + str(self.sim_time ()) + ', I would send an MA
        req now.')

# The final function called at the end of ETCS session
# Its purpose is to calculate final statistics
def summary (self, action, data):
    # Get a handle to the RBC node, which is stores statistics
    rbc_node = self.get_tier_node('RBC')

    # Extract the statistics
    TotalMAreqSent = float (rbc_node.get_state ('TotalMAreqSent'))
    TotalMAreqReceived = float (rbc_node.get_state ('TotalMAreqReceived'))
    TotalMASent = float (rbc_node.get_state ('TotalMASent'))
    TotalMAReceived = float (rbc_node.get_state ('TotalMAReceived'))

    # Calculate the total packet loss in the uplink direction
    self.ETCSlossUPLINK = (TotalMAreqSent - TotalMAreqReceived) /
        TotalMAreqSent
    # Register the result
    stat_handle = rbc_node.stat_register('Network: ETCS msg loss (uplink)', Aps
        .Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
    stat_handle.write ( self.ETCSlossUPLINK )
    # Print a confirmation in the simulation log
    self.sim_message('Time: ' + str(self.sim_time ()) + ', ETCS msg loss (
        uplink) = ' + str(self.ETCSlossUPLINK))

    # Calculate the total packet loss in the downlink direction
```

```
    self.ETCSlossDOWNLINK = (TotalMASent - TotalMAReceived) / TotalMASent
    # Register the result
    stat_handle = rbc_node.stat_register('Network: ETCS msg loss (downlink)',
        Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
    stat_handle.write ( self.ETCSlossDOWNLINK )
    # Print a confirmation in the simulation log
    self.sim_message('Time: ' + str(self.sim_time ()) + ', ETCS msg loss (
        downlink) = ' + str(self.ETCSlossDOWNLINK))

    # Calculate the overall packet loss
    self.ETCSloss = (TotalMAreqSent + TotalMASent - TotalMAreqReceived -
        TotalMAReceived) / (TotalMAreqSent + TotalMASent)
    # Register the result
    stat_handle = rbc_node.stat_register('Network: ETCS msg loss (uplink and
        downlink)', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
    stat_handle.write ( self.ETCSloss )
    # Print a confirmation in the simulation log
    self.sim_message('Time: ' + str(self.sim_time ()) + ', ETCS msg loss (
        overall) = ' + str(self.ETCSloss))
```

Script block no. 1

This script is executed when the *Session Initiation* message is sent by the OBU.

```
# Store retransmission parameters as OBU states
self.state_node = self.get_tier_node ('OBU')
# Set retransmission parameters for the establishment phase
self.state_node.set_state ('etcs_MaxRetransmissions', 20)
self.state_node.set_state ('etcs_RetransmissionTimeout', 1.0)
```

Script block no. 2

This script is executed when the RBC receives the *Session Initiation* message.

```
# Register the parameter as global statistics via the RBC node (for parameter
    studies)
self.app_node = self.get_tier_node('RBC')

# Retransmission parameters
stat_handle = self.app_node.stat_register('Parameter: Max No Retransmissions',
    Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (self.MaxRetransmissions)
stat_handle = self.app_node.stat_register('Parameter: Retransmission Timeout',
    Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (round(self.RetransmissionTimeout, 3))

# ETCS message size
stat_handle = self.app_node.stat_register('Parameter: ETCS message size', Aps.
    Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (round(self.MAsize, 3))

# Find the number of OBUs in the network and register it as a statistic
self.node_list = self.get_nodes_compatible_with_tier('OBU')
stat_handle = self.app_node.stat_register('Parameter: Number of OBUs in the
    network', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (len(self.node_list))
```

Script block no. 3

This script is executed after the OBU receives the *ACK of Train Data* message, i.e. when the ETCS session is successfully established.

```
# Store retransmission parameters as OBU node states. They are used later by UDP retransmission process.
self.obu_node = self.get_tier_node ('OBU')
self.obu_node.set_state ('etcs_MaxRetransmissions', self.MaxRetransmissions)
self.obu_node.set_state ('etcs_RetransmissionTimeout', self.RetransmissionTimeout)

# Schedule Movement Authority requests from this OBU
# for the rest of simulation
OldInterval = 0
for x in range(1, self.MAnumber+1)
    # Calculate when to schedule an MA request
    NextInterval = OldInterval + self.dist_uniform(self.MAinterval*2)
    Schedule = self.schedule_function (NextInterval, self.gen_ma_req, None)
    # Remember when was the last one scheduled
    OldInterval = NextInterval

# Afterwards, schedule a function that will summarize the statistics
self.schedule_function (NextInterval+10, self.summary, None)

# Update the statistic about the number of established ETCS connections
self.rbc_node = self.get_tier_node('RBC')
ETCSconnCounter = self.rbc_node.get_state ('ETCSConnectionsEstablished')
# If there is no ETCSConnectionsEstablished state yet, initiate it.
if ETCSconnCounter == None:
    ETCSconnCounter = 0
# ETCS connection has been established. Increase the counter.
ETCSconnCounter += 1
# Store the counter as RBC state for future
self.rbc_node.set_state ('ETCSConnectionsEstablished', ETCSconnCounter)
# Store the counter value as a global statistic
stat_handle = self.rbc_node.stat_register('Network: ETCS Connections Established', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (ETCSconnCounter)
```

A.2 ETCS application model: Child process responsible for MA update procedure

The child process is called each time an ETCS MA is to be updated. The OBU sends an MA request (including a position report). The RBC replies with an MA grant. This procedure is illustrated in Figure 3.8 on page 45.

Initialization block

This script is executed when this child process is initiated.

```
# Read the parameters passed from the parent process
# ETCS message size
self.msg_size = self.get_parameter ('MAsize')

# Set the MA request message size according to the received value
req_msg = self.get_message ('req')
req_msg.size = self.msg_size

# Similarly, set the MA message size according to the received value
ma_msg = self.get_message ('ma')
ma_msg.size = self.msg_size
```

Script block no. 1

This script is executed when the *Movement Authority Request* is sent by the OBU.

```
# Store the time that the Movement Authority message is sent
self.time_req_sent = self.sim_time()

# Update the statistic about the number of sent MA requests

# Local statistics
self.obu_node = self.get_tier_node('OBU')
MAreqSent = self.obu_node.get_state ('MAreqSent')
# If there is no MAreqSent state yet, initiate it.
if MAreqSent == None:
    MAreqSent = 0
# ETCS MA request has been sent. Increase the counter.
MAreqSent += 1
# Store the counter as OBU state for future
self.obu_node.set_state ('MAreqSent', MAreqSent)
# Store the counter value as a local statistic
stat_handle = self.obu_node.stat_register('ETCS MA requests sent')
stat_handle.write (MAreqSent)

# Global statistics
self.rbc_node = self.get_tier_node('RBC')
TotalMAreqSent = self.rbc_node.get_state ('TotalMAreqSent')
# If there is no TotalMAreqSent state yet, initiate it.
if TotalMAreqSent == None:
    TotalMAreqSent = 0
# ETCS MA request has been sent. Increase the counter.
```

```
TotalMAreqSent += 1
# Store the counter as RBC state for future
self.rbc_node.set_state ('TotalMAreqSent', TotalMAreqSent)
# Store the counter value as a global statistic
stat_handle = self.rbc_node.stat_register('Network: ETCS 1 MA requests sent (
    uplink)', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (TotalMAreqSent)
```

Script block no. 2

This script is executed when the *Movement Authority Request* is received by the RBC.

```
# Check the arrival time, i.e. time that the MA request is received by the RBC
time_req_recevied = self.sim_time()
# Get the RBC node handle
self.rbc_node = self.get_tier_node('RBC')

# Calculate transfer delay and store it as statistics
stat_handle = self.rbc_node.stat_register('ETCS Packet Delay (uplink)', Aps.
    Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (time_req_recevied - self.time_req_sent)
stat_handle = self.rbc_node.stat_register('ETCS Packet Delay (uplink and
    downlink)', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (time_req_recevied - self.time_req_sent)

# Increase the counter of the received MA requests
# and save it as RBC state and as a statistic
TotalMAreqReceived = self.rbc_node.get_state ('TotalMAreqReceived')
# If there is no TotalMAreqSent state yet, then initiate it.
if TotalMAreqReceived == None:
    TotalMAreqReceived = 0
# MA request has been received. Increase the counter.
TotalMAreqReceived += 1
# Store the counter as RBC state for the future use
self.rbc_node.set_state ('TotalMAreqReceived', TotalMAreqReceived)
# Store the counter value as a global statistic
stat_handle = self.rbc_node.stat_register('Network: ETCS 2 MA requests received
     (uplink)', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (TotalMAreqReceived)
```

Script block no. 3

This script is executed when the *Movement Authority* is sent by the RBC.

```
# Store the time that the Movement Authority is sent
self.time_ma_sent = self.sim_time()

# Save the statistics using the RBC node
self.rbc_node = self.get_tier_node('RBC')

# Store the number of MA sent by this RBC
# Get the previous value of the counter, which is stored as an RBC state
TotalMASent = self.rbc_node.get_state ('TotalMASent')
# If there is no TotalMAreqSent state yet, then initiate it.
if TotalMASent == None:
    TotalMASent = 0
```

```
# ETCS MA request has been sent. Increase the counter.
TotalMASent += 1
# Store the counter as RBC state for future use
self.rbc_node.set_state ('TotalMASent', TotalMASent)
# Store the counter value as a global statistic
stat_handle = self.rbc_node.stat_register('Network: ETCS 3 MA sent (downlink)',
      Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (TotalMASent)
```

Script block no. 4

This script is executed when the *Movement Authority* is received by the OBU.

```
# Store the time the Movement Authority is received
time_ma_recevied = self.sim_time()

# Local statistics (at OBU)
self.obu_node = self.get_tier_node('OBU')
stat_handle = self.obu_node.stat_register('ETCS Packet Delay')
stat_handle.write (time_ma_recevied - self.time_ma_sent)
MAreceived = self.obu_node.get_state ('MAreceived')
# If there is no MAreqSent state yet, then initiate it.
if MAreceived == None:
    MAreceived = 0
# ETCS MA request has been sent. Increase the counter.
MAreceived += 1
# Store the counter as OBU state for future
self.obu_node.set_state ('MAreceived', MAreceived)
# Store the counter value as a local statistic
stat_handle = self.obu_node.stat_register('ETCS MA received')
stat_handle.write (MAreceived)

# Global statistics (at RBC)
self.rbc_node = self.get_tier_node('RBC')

# Calculate transfer delay and register it as statistics
stat_handle = self.rbc_node.stat_register('ETCS Packet Delay (downlink)', Aps.
     Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (time_ma_recevied - self.time_ma_sent)
stat_handle = self.rbc_node.stat_register('ETCS Packet Delay (uplink and
     downlink)', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (time_ma_recevied - self.time_ma_sent)

# Increase the received MA state and statistics
TotalMAReceived = self.rbc_node.get_state ('TotalMAReceived')
# If there is no TotalMAreqSent state yet, then initiate it.
if TotalMAReceived == None:
    TotalMAReceived = 0
# MA request has been received. Increase the counter.
TotalMAReceived += 1
# Store the counter as RBC state for the future
self.rbc_node.set_state ('TotalMAReceived', TotalMAReceived)
# Store the counter value as a global statistic
stat_handle = self.rbc_node.stat_register('Network: ETCS 4 MA received (
     downlink)', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (TotalMAReceived)
```

A.3 ETCS application model: Retransmission mechanism

This child process is called to handle each ETCS message sent by the OBU or the RBC. Its purpose is to model the retransmission mechanism at the ETCS layer, which is shown in Figure 3.9 on page 46.

Software implementation of this process is based on *UDP_Drop_Response* process, which is included in the AppTransaction Xpert process library.

Initialization block

This script is executed when the process is initiated.

```
# Get retransmission parameters that were stored as RBC states by the ETCS main
      process
self.state_node = self.get_tier_node ('OBU')
self.max_num_retrans_from_state = self.state_node.get_state ('
     etcs_MaxRetransmissions')
self.timeout_value_from_state = self.state_node.get_state ('
     etcs_RetransmissionTimeout')

self.msg_dict = {}
self.msg_id_dict = {}
self.ack_rcvd_dict = {}
self.debug = 0
```

Function block

```
# This function is executed when a message is received
def msg_receipt_callback_func (self, action, args_tuple):
   msg_id = args_tuple [0]
   tier1 = args_tuple [1]
   tier2 = args_tuple [2]
   msg_size = args_tuple [3]
   send_time = args_tuple [4]

   # Get the message ID and associated robustness information
   orig_id = self.msg_id_dict [str (msg_id)]
   retrans_info = self.msg_dict [str (orig_id)]

   # Send the ACK back to the sender
   # Source tier = tier2 and Destination tier = tier1
   ack_msg = self.create_message (retrans_info.ack_size, tier2, tier1, action,
        1, retrans_info.parent_conn_id)
   ack_args_tuple = args_tuple [0]
   new_args_tuple = msg_id, tier1, tier2, msg_size, send_time
   # Register ACK receipt callback
   self.register_receipt_callback (ack_msg, self.ack_receipt_callback_func,
        args_tuple)
   # Register ACK timer callback
   # First find out how long it took to receive the packet   .
   time_in_flight = self.sim_time () - send_time
```

```
    if time_in_flight < retrans_info.timeout_value:
     self.register_timeout_callback (ack_msg, self.ack_timeout_callback_func,
          new_args_tuple, retrans_info.timeout_value - time_in_flight)
    else:
     self.register_timeout_callback (ack_msg, self.ack_timeout_callback_func,
          new_args_tuple, 0)
    # When the message is received (original or one of the retransmitted ones)
         for the first time
    # execute the end dependencies of the original message
    if retrans_info.execution_resumed != 1:
     retrans_info.execution_resumed = 1   <
     self.execute_child_actions (retrans_info.orig_message, 0)
       if self.debug:
       self.sim_message('Time: ' + str(self.sim_time ()) + '\n   Message reached
             the destination.\n'+ '   Message ID: ' + str (msg_id) + ', Original
             ID: ' + str (orig_id)+ '\n   Sending an ACK with a registered ACK
            receipt callback.' + '\n   Executing child actions.')
    else:
       if self.debug:
          self.sim_message('Time: ' + str(self.sim_time ()) + '\n   Message
               reached the destination.\n'+ '   Message ID: ' + str (msg_id) + '
               , Original ID: ' + str (orig_id)+ '\n   Sending an ACK with a
               registered ACK receipt callback.' + '\n   Child actions have
               already been resumed.')

# This function is executed when an ACK is received
def ack_receipt_callback_func (self, action, ack_args_tuple):
    # Get the message ID and associated robustness information
    msg_id = ack_args_tuple [0]
    orig_id = self.msg_id_dict [str (msg_id)]
    retrans_info = self.msg_dict [str (orig_id)]

    # Mark the original message (for which this is ACK)   as received
    # else ignore this as duplicate ACK
    if retrans_info.message_rcvd != 1:
       retrans_info.message_rcvd = 1
       if self.debug:
          self.sim_message('Time: ' + str(self.sim_time ()) + '\n   ACK received
                for message.\n'+'   Original ID: ' + str (orig_id)+'\n   The
               message is marked as received.')
    else:
       if self.debug:
          self.sim_message('Time: ' + str(self.sim_time ()) +'\n   ACK received
               for message.\n'+'   Original ID: ' + str (orig_id)+'\n   This is
               a duplicate ACK.')

# The following function is executed when a message timer expires
def msg_timeout_callback_func (self, action, args_tuple):
    msg_id = args_tuple [0]
    tier1 = args_tuple [1]
    tier2 = args_tuple [2]
    msg_size = args_tuple [3]

    # Get the message ID and associated robustness information
    orig_id = self.msg_id_dict [str (msg_id)]
    retrans_info = self.msg_dict [str (orig_id)]
    # If any prior retransmissions or the original message corresponding to
    # this message has been received, no processing is needed
    if retrans_info.message_rcvd == 1:
       if self.debug:
          self.sim_message('Time: ' + str(self.sim_time ()) + '\n   Timer
               expired for message.\n'+'   Message ID: ' + str (msg_id) + ',
```

```
            Original ID: ' + str (orig_id)+'\n   Ignoring the timeout since
            one of the message retransmissions has already been received.')
   return

# If the message retransmission count has not been exceeded, retransmit it
# else either continue with the trace or quit (as specified by the user)
if retrans_info.num_retrans >= retrans_info.max_num_retrans:
   # The message retransmission count has been reached
   # check to see if we need to continue with trace
   if retrans_info.cont_with_trace == True:
      # One of the retransmitted messages may have been received
      # and the execution resumed. Check that first
      if retrans_info.execution_resumed == 0:
         retrans_info.execution_resumed = 1
         # Execute the remaining tasks, 0 = AtcC_Action_Stage_End
         self.execute_child_actions (retrans_info.orig_message, 0)
         if self.debug:
            self.sim_message('Time: ' + str(self.sim_time ()) +'\n   Timer
                 expired for message.\n'+'   Message ID: ' + str (msg_id) +
                 ', Original ID: ' + str (orig_id)+'\n   Retransmission
                 count has been reached.'+'\n   Continuing with the trace.')
         return
      else:
         if self.debug:
            self.sim_message('Time: ' + str(self.sim_time ()) +'\n   Timer
                 expired for message.\n'+'   Message ID: ' + str (msg_id) +
                 ', Original ID: ' + str (orig_id)+'\n   This trace has
                 already been resumed.')
         return
   else:
      if self.debug:
         self.sim_message('Time: ' + str(self.sim_time ()) +'\n   Timer
              expired for message.\n'+'   Message ID: ' + str (msg_id) + ',
              Original ID: ' + str (orig_id)+'\n   Aborting the trace as
              configured.')
      self.quit ()
      return

# Message should be retransmitted
if self.debug:
   self.sim_message('Time: ' + str(self.sim_time ()) +'\n   Timer expired
        for message.\n'+'   Message ID: ' + str (msg_id) + ', Original ID: '
         + str (orig_id)+'\n   The message has been retransmitted ' + str (
        retrans_info.num_retrans) + ' times.'+'\n   Retransmitting the
        message.')

# Register the retransmission in the simulation statistics
self.app_node = self.get_tier_node(tier1)
stat_handle = self.app_node.stat_register('ETCS Retransmissions')
stat_handle.write (1)

# Update the statistic about the number of retransmitted ETCS messages
# First, read the previous values of the statistics from the RBC states
self.rbc_node = self.get_tier_node('RBC')
TotalETCSRetransmissions = self.rbc_node.get_state ('
     TotalETCSRetransmissions')
TotalETCSRetransmissionsDownlink = self.rbc_node.get_state ('
     TotalETCSRetransmissionsDownlink')
TotalETCSRetransmissionsUplink = self.rbc_node.get_state ('
     TotalETCSRetransmissionsUplink')

# If the states are not defined yet, then initiate them.
```

```
    if TotalETCSRetransmissions == None:
        TotalETCSRetransmissions = 0
    if TotalETCSRetransmissionsDownlink == None:
        TotalETCSRetransmissionsDownlink = 0
    if TotalETCSRetransmissionsUplink == None:
        TotalETCSRetransmissionsUplink = 0

    # Increase the relevant counters and store values as states & statistics.
    if tier1 == 'RBC':
        # If the retransmission is done by the RBC, then it is a downlink retransmission
        TotalETCSRetransmissionsDownlink += 1
        stat_handle = self.rbc_node.stat_register('Network: ETCS Retransmissions (downlink)', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
        stat_handle.write (TotalETCSRetransmissionsDownlink)
        self.rbc_node.set_state ('TotalETCSRetransmissionsDownlink', TotalETCSRetransmissionsDownlink)
    else:
        # Otherwise it is an uplink retransmission
        TotalETCSRetransmissionsUplink += 1
        stat_handle = self.rbc_node.stat_register('Network: ETCS Retransmissions (uplink)', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
        stat_handle.write (TotalETCSRetransmissionsUplink)
        self.rbc_node.set_state ('TotalETCSRetransmissionsUplink', TotalETCSRetransmissionsUplink)

    # Total number of retransmission, i.e. downlink and uplink sum
    TotalETCSRetransmissions += 1
    stat_handle = self.rbc_node.stat_register('Network: ETCS Retransmissions (uplink and downlink)', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
    stat_handle.write (TotalETCSRetransmissions)
    self.rbc_node.set_state ('TotalETCSRetransmissions', TotalETCSRetransmissions)

    # Increment the number of retransmissions done for this message
    retrans_info.num_retrans = retrans_info.num_retrans + 1
    # RETRANSMIT THE MESSAGE - Create the message, associate the same retrans_info
    retrans_msg = self.create_message (msg_size, tier1, tier2, action, 1, retrans_info.parent_conn_id)

    # Associate the retransmission information with the message
    self.msg_id_dict [str (retrans_msg.id)] = retrans_info.orig_message_id
    retrans_args_tuple = retrans_msg.id, retrans_msg.start_tier, retrans_msg.end_tier, retrans_msg.size, self.sim_time ()
    # Register for timeout and receipt callback for this message as well
    self.register_receipt_callback (retrans_msg, self.msg_receipt_callback_func, retrans_args_tuple)
    self.register_timeout_callback (retrans_msg, self.msg_timeout_callback_func, retrans_args_tuple, retrans_info.timeout_value)

# This function is executed when an ACK timer expires
def ack_timeout_callback_func (self, action, ack_args_tuple):
    self.msg_timeout_callback_func (action, ack_args_tuple)
    # No action is needed, the callback is needed in order for the ACE brain
    # to account for possible loss messages
    if self.debug:
        orig_id = ack_args_tuple [0]
        retrans_info = self.msg_dict [str (orig_id)]
        self.sim_message('Time: ' + str(self.sim_time ()) +'\n   Timer expired for an ACK.\n'+'   Original ID: ' + str (orig_id))
```

Script block no. 1

This script is executed whenever one of the ETCS nodes (i.e. application tiers) sends a message.

```
# Get the message marked as "reliable" in this WB
message = self.get_message ("reliable")

# Get the original message
orig_message = self.get_parameter ("Original Message")
# Get message size from the original message
message.size = orig_message.size

# Create python object that will store the robustness information
retrans_info = app_robustness_ack.app_robustness_ack (orig_message)

# Get the parameter values as specified by the user in ETCS task configuration
retrans_info.max_num_retrans = self.max_num_retrans_from_state
retrans_info.timeout_value = self.timeout_value_from_state

# Get the parameters passed by the parent process
retrans_info.ack_size = self.get_parameter ("Size of Acknowledgement")
retrans_info.parent_conn_id = self.get_parameter ("Parent Connection ID")
retrans_info.cont_with_trace = self.get_parameter ("Continue with Trace")
self.debug = self.get_parameter ("Print Debug Information")

# Store the ID of the original message
retrans_info.orig_message_id = orig_message.id
# Make sure the "reliable" message uses the same connection index as the
    message in the parent WB
self.set_action_conn_index (message, 1)
# Make sure that the connection index is not retrieved from the child task
self.set_action_conn_index_override (message)
# Store the Original Message ID and the Robustness info in two different
    dictionaries for later use
self.msg_id_dict [str (message.id)] = retrans_info.orig_message_id
self.msg_dict [str (orig_message.id)] = retrans_info

# Create argument tuple that will be associated with timeout and receipt
    callbacks
args_tuple = message.id, message.start_tier, message.end_tier, message.size,
    self.sim_time ()
# Register a receipt and timeout callback
self.register_receipt_callback (message, self.msg_receipt_callback_func,
    args_tuple)
self.register_timeout_callback (message, self.msg_timeout_callback_func,
    args_tuple, retrans_info.timeout_value)

# Print debug information
if self.debug:
    self.sim_message('Time: ' + str(self.sim_time ()) + '\n   Message sent. '
           '\n   Message ID: '  + str(message.id) + ', Timeout value: '+ str(
        retrans_info.timeout_value) + ', Retransmission count: ' + str (
        retrans_info.max_num_retrans))
```

A.4 VoLTE one-to-one call model: Signalling plane process

This process models VoLTE one-to-one call setup procedure, i.e. the signalling plane of the call. Figure 4.2 on page 94 illustrates the detailed SIP message exchange during the procedure, while Figure 4.5 on page 97 shows this message exchange implemented in OPNET ATX.

Initialization block

This script is executed when the process is initiated.

```
# Read the average call duration from the parameters
self.CallDuration = self.get_parameter ('CallDuration')
# Check if the value is valid, quit if not
if self.CallDuration <= 0:
    self.sim_message ('The value for parameter "CallDuration" should be greater
          than 0.', 'Quitting the task.')
    self.quit ()

# Read the number of voice frames per second from the parameters
self.VoiceFrames = self.get_parameter ('VoiceFrames')
# Check if the value is valid, quit if not
if self.VoiceFrames <= 0:
    self.sim_message ('The value for parameter "VoiceFrames" should be greater
         than 0.', 'Quitting the task.')
    self.quit ()

# Read the voice frame size from the parameters
self.VoiceFrameSize = self.get_parameter ('VoiceFrameSize')
# Check if the value is valid, quit if not
if self.VoiceFrameSize <= 0:
    self.sim_message ('The value for parameter "VoiceFrameSize" should be
         greater than 0.', 'Quitting the task.')
    self.quit ()
```

Script block no. 1

This script is executed when the *SIP INVITE* message is sent by the call initiator.

```
# Store the initiation time
self.time_req_sent = self.sim_time()

# Update the statistic about the number of initiated calls

# Local at the Terminal node
self.terminal_node = self.get_tier_node('Terminal')
OneInitiatedCalls = self.terminal_node.get_state ('OneInitiatedCalls')

# If there is no OneInitiatedCalls state yet, initiate it.
if OneInitiatedCalls == None:
    OneInitiatedCalls = 0

```

```
# SIP INVITE has been sent. Increase the counter.
OneInitiatedCalls += 1
# Store the counter as a node state for future use
self.terminal_node.set_state ('OneInitiatedCalls', OneInitiatedCalls)
# Store the counter value as a local statistic
stat_handle = self.terminal_node.stat_register('One2One: Initiated calls')
stat_handle.write (OneInitiatedCalls)
```

Script block no. 2

This script is executed when the *SIP INVITE* message is received by the CSCF.

```
# Global at the CSCF node
self.cscf_node = self.get_tier_node('CSCF')
OneTotalInitiatedCalls = self.cscf_node.get_state ('OneTotalInitiatedCalls')

# If there is no OneTotalInitiatedCalls state yet, initiate it.
if OneTotalInitiatedCalls == None:
    OneTotalInitiatedCalls = 0

# SIP INVITE has been sent. Increase the counter.
OneTotalInitiatedCalls += 1
# Store the counter as a node state for future use
self.cscf_node.set_state ('OneTotalInitiatedCalls', OneTotalInitiatedCalls)
# Store the counter value as a global statistic
stat_handle = self.cscf_node.stat_register('One2One: Total initiated calls',
     Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (OneTotalInitiatedCalls)

stat_handle = self.cscf_node.stat_register('Parameter: One2One Voice frames per
      second', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (self.VoiceFrames)

stat_handle = self.cscf_node.stat_register('Parameter: One2One Voice frame size
      (bytes)', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (self.VoiceFrameSize)
```

Script block no. 3

This script is executed when the *SIP RINGING* message is received by the call initiator. This means that the called terminal waits for the user to answer the incoming call.

```
# Check the time
time_ringing = self.sim_time()

# Get the CSCF node handle
self.cscf_node = self.get_tier_node('CSCF')

# Store the call initialization time as a global statistic (via CSCF node)
stat_handle = self.cscf_node.stat_register('One2One: Initialization time (until
      ringing) (s)', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (time_ringing - self.time_req_sent)
```

Script block no. 4

This script is executed when the *SIP ACK* message is received by the called party. This means that the call is established and the voice transmission can begin.

```
# Call has been set up
# Check the arrival time (including "user answer delay")
time_req_recevied = self.sim_time()

# Get the CSCF node handle
self.cscf_node = self.get_tier_node('CSCF')

# Store the initialization time (including user delay)
stat_handle = self.cscf_node.stat_register('One2One: Initialization time (s)',
     Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (time_req_recevied - self.time_req_sent)

# Increase the received Established Calls state and statistics
OneTotalEstablishedCalls = self.cscf_node.get_state ('OneTotalEstablishedCalls'
     )

# If there is no OneTotalEstablishedCalls state yet, initiate it.
if OneTotalEstablishedCalls == None:
    OneTotalEstablishedCalls = 0

# One to one call has been established. Increase the counter.
OneTotalEstablishedCalls += 1
# Store the counter as CSCF state for the future use
self.cscf_node.set_state ('OneTotalEstablishedCalls', OneTotalEstablishedCalls)
# Store the counter value as a global statistic
stat_handle = self.cscf_node.stat_register('One2One: Total established calls',
     Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (OneTotalEstablishedCalls)

# Local at the Destination node
self.dest_node = self.get_tier_node('Destination')
OneEstablishedCalls = self.dest_node.get_state ('OneEstablishedCalls')

# If there is no OneInitiatedCalls state yet. Thus, initiate it.
if OneEstablishedCalls == None:
    OneEstablishedCalls = 0

# ETCS MA req has been sent. Increase the counter.
OneEstablishedCalls += 1
# Store the counter as a node state for future use
self.dest_node.set_state ('OneEstablishedCalls', OneEstablishedCalls)
# Store the counter value as a local statistic
stat_handle = self.dest_node.stat_register('One2One: Received calls')
stat_handle.write (OneEstablishedCalls)

# Determine call duration based on the average call duration
self.ThisCallDuration = int(self.dist_uniform(self.CallDuration*2))
# Call has to last at least 5 seconds
if self.ThisCallDuration < 5:
    self.ThisCallDuration = 5

# Store the call duration value as a global statistic
stat_handle = self.cscf_node.stat_register('One2One: Call Duration (s)', Aps.
     Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (self.ThisCallDuration)

```

```
# Pass the call duration value to the child task
child_params = {'CallDuration':self.ThisCallDuration,'VoiceFrames':self.
    VoiceFrames,'VoiceFrameSize':self.VoiceFrameSize}
# Initiate child task - RTP media stream
self.invoke_child_task(action, 'MediaStream.aed.m', True, child_params)
```

Script block no. 5

This script is executed when the final *SIP* message is received by the called party and the call is finished.

```
self.sim_message('Time: ' + str(self.sim_time ()) + ', Call has successfully
    ended.')
```

A.5 VoLTE one-to-one call model: Media plane process

This process models the media plane of a VoLTE one-to-one call. All of the messages in this process—packets carrying voice samples—are generated and then handled by functions defined in the *Function Block*.

Initialization block

This script is executed when the process is initiated.

```
# Read the call duration from the parameters passed by the parent process
self.CallDuration = self.get_parameter('CallDuration')

# Schedule when to send the last message that will end this child process
bye_msg = self.get_message('EndMsg')
bye_msg.tier_delay = self.CallDuration + 10.0

# Read parameters passed from the parent process
self.VoiceFrameSize = self.get_parameter('VoiceFrameSize')
VoiceFrames = self.get_parameter('VoiceFrames')

# Calculate the time period between voice frames
interframe = float(1.0/VoiceFrames)
# Calculate how many voice frames will be sent during the call
NoVoiceFrames = int(self.CallDuration * VoiceFrames)

# Schedule all RTP voice packets, which carry the voice frames, for the whole
     call duration
OldInterval = 0
for x in range(1, NoVoiceFrames):
    # Calculate when to schedule the next RTP voice packet
    NextInterval = OldInterval + interframe
    # Schedule a function, which will send the packet
    self.schedule_function (NextInterval, self.SendVoicePacket, None)
    # Store the interval value for the next iteration
    OldInterval = NextInterval

# Schedule a function, which will calculate final statistics at the end of the
     call
self.schedule_function (OldInterval + 5, self.CalculateLoss, None)
```

Function block

```
# Function sending RTP packets
def SendVoicePacket (self, action, data):
    # Store the time when the packet is sent
    time_rtp_sent = self.sim_time()

    # Send an RTP packet from the terminal towards the destination
    rtp_packet_1 = self.create_message(self.VoiceFrameSize, "Terminal", "
         Destination")
    # Register a callback function at the receiver
```

```
    self.register_receipt_callback(rtp_packet_1, self.ReceiveRTPPacket, ("Destination",time_rtp_sent))
    self.register_timeout_callback(rtp_packet_1, self.Ignore, {}, 1.0)

    # Send an RTP packet from the destination towards the terminal
    rtp_packet_2 = self.create_message(self.VoiceFrameSize, "Destination", "Terminal")
    # Register a callback function at the receiver
    self.register_receipt_callback(rtp_packet_2, self.ReceiveRTPPacket, ("Terminal",time_rtp_sent))
    self.register_timeout_callback(rtp_packet_2, self.Ignore, {}, 1.0)

    # Update statistics
    terminal_node = self.get_tier_node('Terminal')
    destiantion_node = self.get_tier_node('Destination')
    cscf_node = self.get_tier_node('CSCF')

    OnePacketsSent = terminal_node.get_state ('OnePacketsSent')
    OnePacketsSent2 = destiantion_node.get_state ('OnePacketsSent')
    OneTotalPacketsSent = cscf_node.get_state ('OneTotalPacketsSent')

    # If there is no OnePacketsSent state yet, initiate it.
    if OnePacketsSent == None:
        OnePacketsSent = 0
    if OnePacketsSent2 == None:
        OnePacketsSent2 = 0
    if OneTotalPacketsSent == None:
        OneTotalPacketsSent = 0

    # RTP packets have been sent. Increase the counters.
    OnePacketsSent += 1
    OnePacketsSent2 += 1
    OneTotalPacketsSent += 2 # Total counter is increased by two

    # Store the counter as node states for future use
    terminal_node.set_state ('OnePacketsSent', OnePacketsSent)
    destiantion_node.set_state ('OnePacketsSent', OnePacketsSent2)
    cscf_node.set_state ('OneTotalPacketsSent', OneTotalPacketsSent)

    # Store the counter values as local statistics
    stat_handle = terminal_node.stat_register('One2One: RTP voice packets sent')
    stat_handle.write (OnePacketsSent)

    stat_handle = destiantion_node.stat_register('One2One: RTP voice packets sent')
    stat_handle.write (OnePacketsSent2)

    stat_handle = cscf_node.stat_register('One2One: Total RTP voice packets sent', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
    stat_handle.write (OneTotalPacketsSent)

# Function receiving RTP packets
def ReceiveRTPPacket (self, action, data):
    # Read arguments passed to the function
    receiver, time_rtp_sent = data

    # Check the arrival time
    time_rtp_received = self.sim_time()
    # Calculate packet delay
    delay = time_rtp_received - time_rtp_sent
```

```
        # Get the receiver node (terminal/destination) & CSCF node handles
        receiver_node = self.get_tier_node(receiver)
        cscf_node = self.get_tier_node('CSCF')

        # Store the voice packet delay as a global statistic
        stat_handle = cscf_node.stat_register('One2One: RTP voice packet delay (s)', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
        stat_handle.write (delay)

        # Store the voice packet delay as a global statistic
        stat_handle = receiver_node.stat_register('One2One: RTP voice packet delay (s)')
        stat_handle.write (delay)

        # Jitter calculations
        PreviousDelay = receiver_node.get_state ('PreviousDelay')
        PreviousJitter = receiver_node.get_state ('Jitter')

        if PreviousDelay == None:
            # It is the first sample, initialize the arguments
            jitter = 0.0
        else:
            d = abs(delay - PreviousDelay)
            jitter = float(PreviousJitter + ((d - PreviousJitter) / 16))
            stat_handle = receiver_node.stat_register('One2One: RTP jitter')
            stat_handle.write (jitter)
            stat_handle = cscf_node.stat_register('One2One: RTP jitter', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
            stat_handle.write (jitter)

        receiver_node.set_state ('PreviousDelay', delay)
        receiver_node.set_state ('Jitter', jitter)

        # Local: Number of received RTP packets
        OnePacketsReceived = receiver_node.get_state ('OnePacketsReceived')
        # If there is no OnePacketsReceived state yet. Thus, initiate it.
        if OnePacketsReceived == None:
            OnePacketsReceived = 0

        # RTP packet has been received. Increase the counter.
        OnePacketsReceived += 1
        # Store the counter as a node state for future use
        receiver_node.set_state ('OnePacketsReceived', OnePacketsReceived)
        # Store the counter value as a local statistic
        stat_handle = receiver_node.stat_register('One2One: RTP voice packets received')
        stat_handle.write (OnePacketsReceived)

        # Global: Number of received RTP packets
        OneTotalPacketsReceived = cscf_node.get_state ('OneTotalPacketsReceived')
        #If there is no OnePacketsReceived state yet. Thus, initiate it.
        if OneTotalPacketsReceived == None:
            OneTotalPacketsReceived = 0

        # RTP packet has been received. Increase the counter.
        OneTotalPacketsReceived += 1
        # Store the counter as a node state for future use
        cscf_node.set_state ('OneTotalPacketsReceived', OneTotalPacketsReceived)
        # Store the counter value as a local statistic
        stat_handle = cscf_node.stat_register('One2One: Total RTP voice packets received', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
        stat_handle.write (OneTotalPacketsReceived)
```

```
# Function calculating the global performance measures
def CalculateLoss (self, action, data):
    # Access the CSCF node
    cscf_node = self.get_tier_node('CSCF')

    # Read the states (number of received/send packets)
    OneTotalPacketsSent = float(cscf_node.get_state ('OneTotalPacketsSent'))
    OneTotalPacketsReceived = float(cscf_node.get_state ('
        OneTotalPacketsReceived'))

    # Calculate the packet loss
    RTPpacketLoss = float((OneTotalPacketsSent - OneTotalPacketsReceived) /
        OneTotalPacketsSent)
    # Store the packet loss as a statistic
    stat_handle = cscf_node.stat_register('One2One: RTP voice packet loss', Aps
        .Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
    stat_handle.write (RTPpacketLoss)

def Ignore (self, action, data):
    Empty = 0
```

A.6 VoLTE REC model: Signalling plane process

This process models VoLTE REC setup procedure, i.e. the signalling plane of the call. Figure 4.6 on page 97 shows the SIP message exchange of this setup procedure implemented in OPNET ATX.

Initialization block

This script is executed when the process is initiated.

```
# Read the average call duration from the parameters
CallDuration = self.get_parameter ('CallDuration')
# Check is the value is valid, quit if not
if CallDuration <= 0:
    self.sim_message ('The value for parameter "CallDuration" should be greater
          than 0.', 'Quitting the task.')
    self.quit ()

# Determine call duration based on the average call duration
self.ThisCallDuration = int(self.dist_uniform(CallDuration*2))
# Call has to last at least 5 seconds
if self.ThisCallDuration < 5:
    self.ThisCallDuration = 5

self.VoiceFrames = self.get_parameter ('VoiceFrames')
# Check is the value is valid, quit if not
if self.VoiceFrames <= 0:
    self.sim_message ('The value for parameter "VoiceFrames" should be greater
         than 0.', 'Quitting the task.')
    self.quit ()

self.VoiceFrameSize = self.get_parameter ('VoiceFrameSize')
# Check is the value is valid, quit if not
if self.VoiceFrameSize <= 0:
    self.sim_message ('The value for parameter "VoiceFrameSize" should be
         greater than 0.', 'Quitting the task.')
    self.quit ()
```

Script block no. 1

This script is executed when the *SIP INVITE* message is sent by the call initiator.

```
# Store the initiation time
self.time_req_sent = self.sim_time()

self.terminal_node = self.get_tier_node('Terminal')

# Update the local statistic about the number of initiated calls at the
     Terminal node
RECInitiatedCalls = self.terminal_node.get_state ('RECInitiatedCalls')
# If there is no RECInitiatedCalls state yet, initiate it.
if RECInitiatedCalls == None:
    RECInitiatedCalls = 0
# SIP INVITE has been sent. Increase the counter.
RECInitiatedCalls += 1
```

```
# Store the counter as a node state for future use
self.terminal_node.set_state ('RECInitiatedCalls', RECInitiatedCalls)
# Store the counter value as a local statistic
stat_handle = self.terminal_node.stat_register('REC: Initiated calls')
stat_handle.write (RECInitiatedCalls)
```

Script block no. 2

This script is executed when the *SIP INVITE* message is received by the CSCF.

```
# Get handle to the CSCF node, which is used to store statistics
self.cscf_node = self.get_tier_node('CSCF')

# Store the call duration value as a global statistic
stat_handle = self.cscf_node.stat_register('REC: Call Duration (s)', Aps.
     Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (self.ThisCallDuration)

# Global at the CSCF node
RECTotalInitiatedCalls = self.cscf_node.get_state ('RECTotalInitiatedCalls')
#If there is no RECTotalInitiatedCalls state yet, initiate it.
if RECTotalInitiatedCalls == None:
    RECTotalInitiatedCalls = 0
# SIP INVITE has been sent. Increase the counter.
RECTotalInitiatedCalls += 1
# Store the counter as a node state for future use
self.cscf_node.set_state ('RECTotalInitiatedCalls', RECTotalInitiatedCalls)
# Store the counter value as a global statistic
stat_handle = self.cscf_node.stat_register('REC: Total initiated calls', Aps.
     Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (RECTotalInitiatedCalls)

stat_handle = self.cscf_node.stat_register('Parameter: REC Voice frames per
     second', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (self.VoiceFrames)

stat_handle = self.cscf_node.stat_register('Parameter: REC Voice frame size (
     bytes)', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (self.VoiceFrameSize)
```

Script block no. 3

This script is executed when the first two calls—between the call initiator and the RSAP and between the RSAP and the Control Centre (i.e. the train dispatcher)—are initialized. At this time, the RSAP can initiate calls to the listening nodes.

```
# Get node handles
self.controlcenter_node = self.get_tier_node('ControlCenter')
self.rsap_node = self.get_tier_node('RSAP')

criteria_dict = {'model':'lte_wkstn_adv'}
DestinationNodes = self.get_nodes(criteria_dict)
self.sim_message('Time: ' + str(self.sim_time ()) + ' Number of potential
     listening nodes: %d' % (len(DestinationNodes)))

```

```
i = 0

# List with node names
ListOfListeners = list()
# List with tier names, used later to send media. It is kept empty until the
    call setup procedure is done at each listening tier.
self.ListOfListeningTiers = list()
# Remember
self.rsap_node.set_state ('ListOfListeningTiers', self.ListOfListeningTiers)

terminal_node_name = self.terminal_node.get_attr('name', 'Unknown')
rsap_node_name = self.rsap_node.get_attr('name', 'Unknown')
controlcenter_node_name = self.controlcenter_node.get_attr('name', 'Unknown')
cscf_node_name = self.cscf_node.get_attr('name', 'Unknown')

for item in DestinationNodes:
    item_node_name = item.get_attr('name', 'Unknown')
    if (item_node_name == terminal_node_name):
        pass
    elif (item_node_name == rsap_node_name):
        pass
    elif (item_node_name == controlcenter_node_name):
        pass
    elif (item_node_name == cscf_node_name):
        pass
    elif not(item_node_name in ListOfListeners) and (('client' in
         item_node_name.lower()) or ('ue' in item_node_name.lower()) or ('train
         ' in item_node_name.lower())):
        i += 1
        ListOfListeners.append(item_node_name)
        tier_name = 'List' + str(i)
        child_params = {'CallDuration':self.ThisCallDuration, 'ListeningNode':
             item_node_name, 'TierName':tier_name}
        tier_map = {'Listener':tier_name}
        self.invoke_child_task(action, 'RECCalltoListeningNodes1.aed.m', False,
              child_params, tier_map)
    else:
        pass

# Store the number of nodes value as a global statistic
stat_handle = self.cscf_node.stat_register('REC: Listening nodes', Aps.
     Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (len(ListOfListeners))

self.sim_message('Number of listening nodes detected: %d' % (len(
     ListOfListeners)))
```

Script block no. 4

This script is executed when the *SIP ACK* message is received by the RSAP. This means that the call is established and the voice transmission between the call initiator and the Control Centre (i.e. the train dispatcher) can begin.

```
# The Railway Emergency Call has been set up
# Check the arrival time of this message
time_req_recevied = self.sim_time()

# Calculate the initialization time
```

```
# and store it in a relevant statistic
stat_handle = self.cscf_node.stat_register('REC: Initialization time (s)', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (time_req_recevied - self.time_req_sent)

# Increase the received Established Calls state and statistic
RECTotalEstablishedCalls = self.cscf_node.get_state ('RECTotalEstablishedCalls')
# If there is no RECTotalEstablishedCalls state yet, initiate it.
if RECTotalEstablishedCalls == None:
    RECTotalEstablishedCalls = 0
# REC call has been established. Increase the counter.
RECTotalEstablishedCalls += 1
# Store the counter as CSCF state for the future use
self.cscf_node.set_state ('RECTotalEstablishedCalls', RECTotalEstablishedCalls)
# Store the counter value as a global statistic
stat_handle = self.cscf_node.stat_register('REC: Total established calls', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
stat_handle.write (RECTotalEstablishedCalls)

# Pass the call duration value to the child task
child_params = {'CallDuration':self.ThisCallDuration,'VoiceFrames':self.VoiceFrames,'VoiceFrameSize':self.VoiceFrameSize}
# Initiate a child task - RTP media stream
self.invoke_child_task(action, 'RECMediaStream.aed.m', True, child_params)
```

Script block no. 5

This script is executed when the final *SIP* message is received and the call is finished.

```
self.sim_message('Time: ' + str(self.sim_time ()) + ', Call has successfully ended.')
```

A.7 VoLTE REC model: Child process responsible for the signalling exchange with a listening node

This child process models the SIP signalling exchange between the RSAP and a listening node. The process is called for each of the listening nodes in the network.

Initialization block

This script is executed when the process is initiated.

```
# Get the name of the particular listening node for which this process was called
self.DestinationNodeName = str(self.get_parameter('ListeningNode'))
self.TierName = str(self.get_parameter('TierName'))
# Set the listening node as the tier node for this child process
self.set_tier_node(self.TierName, self.DestinationNodeName)

# Read the average call duration from the parameters
self.CallDuration = self.get_parameter('CallDuration')
# Schedule the final message, which will close this child process
# Add a 5 s buffer time at the end for possible delays in the network
bye_msg = self.get_message('End')
bye_msg.tier_delay = self.CallDuration + 5.0

# Schedule start of SIP call establishment procedure
self.schedule_function (0.0005, self.SendSIPinvite, None)

# Print a confirmation message in the simulation log
self.sim_message('Time: ' + str(self.sim_time ()) + ' Child task initialized')
```

Function block

```
# Function used to send the first SIP INVITE to the listening node
def SendSIPinvite (self, action, data):
    # Store the time when the INVITE packet is sent
    self.time_invite_sent = self.sim_time()

    # Send a SIP packet from the RSAP towards CSCF
    step = 1
    sip_packet = self.create_message(600, 'RSAP', 'CSCF')
    # Register a callback function at the receiver
    # Including the step number and the listening node name (TierName) as the destination
    self.register_receipt_callback(sip_packet, self.CSCFforward, (self.TierName, step))

# Function called at the CSCF in the event of receiving a SIP message
def CSCFforward (self, action, data):
    # Read arguments passed to the function
    destination, step = data

    # Depending on the direction, forward the message to the RSAP or to the listening node
    if (destination == 'RSAP'):
```

```
        sip_packet = self.create_message(600, 'CSCF', 'RSAP')
        self.register_receipt_callback(sip_packet, self.ReceiveSIPPacket, ('RSAP', step))
    elif (destination == self.TierName):
        sip_packet = self.create_message(600, 'CSCF', self.TierName)
        self.register_receipt_callback(sip_packet, self.ReceiveSIPPacket, (self.TierName, step))

# Function called at the destination node in the event of receiving an incoming SIP message
def ReceiveSIPPacket (self, action, data):
    destination, step = data

    # If step 9 is reached, then the call has been successfully established
    if step == 9:
        # Schedule SendSIPbye function, which will generate SIP signalling
        # at the end of the call
        self.schedule_function (self.CallDuration+0.0005, self.SendSIPbye, None)

        # Get handles to the RSAP node in order to update the list of listening nodes
        rsap_node = self.get_tier_node('RSAP')
        # Add the just established call to the list of the listening nodes
        ListOfListeningTiers = rsap_node.get_state ('ListOfListeningTiers')
        ListOfListeningTiers.append(self.TierName)
        # Save the list as an RSAP state
        rsap_node.set_state ('ListOfListeningTiers', ListOfListeningTiers)

        # Local statistics at the destination node
        dest_node = self.get_tier_node(self.TierName)
        RECEstablishedListener = dest_node.get_state ('RECEstablishedListener')
        # If there is no RECEstablishedListener state yet, initiate it.
        if RECEstablishedListener == None:
            RECEstablishedListener = 0
        # SIP INVITE has been sent. Increase the counter.
        RECEstablishedListener += 1
        # Store the counter as a node state for future use
        dest_node.set_state ('RECEstablishedListener', RECEstablishedListener)
        # Store the counter value as a local statistic
        stat_handle = dest_node.stat_register('REC: Received listening calls')
        stat_handle.write (RECEstablishedListener)

        # Global statistics - saved in the CSCF node
        cscf_node = self.get_tier_node('CSCF')
        RECTotalEstablishedListeners = cscf_node.get_state ('RECTotalEstablishedListeners')
        # If there is no RECTotalEstablishedListeners state yet, initiate it.
        if RECTotalEstablishedListeners == None:
            RECTotalEstablishedListeners = 0
        # SIP INVITE has been sent. Increase the counter.
        RECTotalEstablishedListeners += 1
        # Store the counter as a node state for future use
        cscf_node.set_state ('RECTotalEstablishedListeners', RECTotalEstablishedListeners)
        # Store the counter value as a local statistic
        stat_handle = cscf_node.stat_register('REC: Established listening calls', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
        stat_handle.write (RECTotalEstablishedListeners)

    # The last step - the SIP call is finished
    elif step == 11:
```

```
        # End of the process
        pass

    # If none of the above, create a SIP message and send it as a reply
    else:
        # Increase the counter
        step += 1
        if (destination == 'RSAP'):
            # Create a SIP reply
            sip_packet = self.create_message(600, 'RSAP', 'CSCF')
            # Register a callback function at CSCF node
            self.register_receipt_callback(sip_packet, self.CSCFforward, (self.TierName, step) )
        elif (destination == self.TierName):
            # Create a SIP reply
            sip_packet = self.create_message(600, self.TierName, 'CSCF')
            # Register a callback function at CSCF node
            self.register_receipt_callback(sip_packet, self.CSCFforward, ('RSAP', step) )

# Initiate the final SIP message exchange that will finish the call
def SendSIPbye(self, action, data):
    # Send a SIP packet from the terminal towards CSCF
    step = 10
    sip_packet = self.create_message(600, 'RSAP', 'CSCF')
    # Register a callback function at the receiver
    self.register_receipt_callback(sip_packet, self.CSCFforward, (self.TierName, step))
```

A.8 VoLTE REC model: Media plane process

This child process models the media plane of a VoLTE REC call. All of the messages in this process—packets carrying voice samples—are generated and then handled by functions included in the *Function Block*.

Initialization block

This script is executed when the process is initiated.

```
# Read the call duration from the parameters passed by the parent process
self.CallDuration = self.get_parameter('CallDuration')

# Schedule when to send the last message that will end this child process
bye_msg = self.get_message('BYEafterMedia')
bye_msg.tier_delay = self.CallDuration + 10.0

# Read parameters passed from the parent process
self.VoiceFrameSize = self.get_parameter('VoiceFrameSize')
VoiceFrames = self.get_parameter('VoiceFrames')

# Get node handles
self.rsap_node = self.get_tier_node('RSAP')
self.terminal_node = self.get_tier_node('Terminal')
self.rsap_node = self.get_tier_node('RSAP')
self.control_node = self.get_tier_node('ControlCenter')

self.TierList = self.rsap_node.get_state ('ListOfListeningTiers')

# Calculate the time period between voice frames
interframe = float(1.0/VoiceFrames)
# Calculate how many voice frames will be sent during the call
NoVoiceFrames = int(self.CallDuration * VoiceFrames)

# Schedule all RTP voice packets, which carry the voice frames, for the whole
    call duration
OldInterval = 0
for x in range(1, NoVoiceFrames):
    # Calculate when to schedule the next RTP voice packet
    NextInterval = OldInterval + interframe
    # Schedule a function, which will send the packet
    self.schedule_function(NextInterval, self.SendVoicePacket, None)
    # Store the interval value for the next iteration
    OldInterval = NextInterval

# Schedule a function, which will calculate final statistics at the end of the
    call
self.schedule_function (OldInterval + 5, self.CalculateLoss, None)

# Voice packet counter
self.packet_counter = 0
```

Function block

```
# Function responsible for sending RTP packets (i.e. voice packets)
# One packet is sent from the terminal towards RSAP, and another is sent from the ControlCenter (i.e. the train dispatcher) also towards RSAP
def SendVoicePacket (self, action, data):
    self.packet_counter += 1

    # Store the time when the packet is sent
    time_rtp_sent = self.sim_time()

    # Send an RTP packet from the terminal towards RSAP
    rtp_packet_1 = self.create_message(self.VoiceFrameSize, 'Terminal','RSAP')
    # Register a callback function at the receiver
    self.register_receipt_callback(rtp_packet_1, self.RSAPdistributeRTP, ('Terminal',time_rtp_sent))

    # Send an RTP packet from the Control Center towards the RSAP
    rtp_packet_2 = self.create_message(self.VoiceFrameSize, 'ControlCenter', 'RSAP')
    # Register a callback function at the receiver
    self.register_receipt_callback(rtp_packet_2, self.RSAPdistributeRTP, ('ControlCenter',time_rtp_sent))

    # Update statistics
    RECPacketsSent = self.terminal_node.get_state ('RECPacketsSent')
    RECPacketsSent2 = self.control_node.get_state ('RECPacketsSent')
    RECTotalPacketsSent = self.rsap_node.get_state ('RECTotalPacketsSent')

    # If there is no OnePacketsSent state yet, initiate it.
    if RECPacketsSent == None:
        RECPacketsSent = 0
    if RECPacketsSent2 == None:
        RECPacketsSent2 = 0
    if RECTotalPacketsSent == None:
        RECTotalPacketsSent = 0

    # RTP packets have been sent. Increase the counters.
    RECPacketsSent += 1
    RECPacketsSent2 += 1
    RECTotalPacketsSent += 2 # Total counter is increased by two

    # Store the counter as node states for future use
    self.terminal_node.set_state ('RECPacketsSent', RECPacketsSent)
    self.control_node.set_state ('RECPacketsSent', RECPacketsSent2)
    self.rsap_node.set_state ('RECTotalPacketsSent', RECTotalPacketsSent)

    # Store the counter values as local statistics
    stat_handle = self.terminal_node.stat_register('REC: RTP voice packets sent')
    stat_handle.write (RECPacketsSent)

    stat_handle = self.control_node.stat_register('REC: RTP voice packets sent')
    stat_handle.write (RECPacketsSent2)

    stat_handle = self.rsap_node.stat_register('REC: Total RTP voice packets sent', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
    stat_handle.write (RECTotalPacketsSent)

# Function called by the RSAP node after receiving an RTP packet
```

```
# RSAP is responsible for forwarding the RTP packet to all of the listening nodes
def RSAPdistributeRTP (self, action, data):
    # Read arguments passed to the function
    sender, time_rtp_sent = data

    # Check which node is the sender
    if (sender == 'Terminal'):
        # If the packet was sent by the terminal (i.e. REC initiator), then forward the packet towards ControlCenter
        rtp_packet = self.create_message(self.VoiceFrameSize, 'RSAP','ControlCenter')
        self.register_receipt_callback(rtp_packet, self.ReceiveRTPPacket, ('ControlCenter', time_rtp_sent))
    elif (sender == 'ControlCenter'):
        # If the packet was sent by the ControlCenter, then forward the packet towards the terminal
        rtp_packet = self.create_message(self.VoiceFrameSize, 'RSAP','Terminal')
        self.register_receipt_callback(rtp_packet, self.ReceiveRTPPacket, ('Terminal', time_rtp_sent))

    # Get the list of listening nodes (the list is checked every 100 frames)
    if ((self.packet_counter % 100)==0) or (self.TierList == None): # Do not check the list if it is empty
        self.TierList = self.rsap_node.get_state ('ListOfListeningTiers')

    # Distribute the packet to all of the listeners
    for node in self.TierList:
        rtp_packet_listen = self.create_message(self.VoiceFrameSize, 'RSAP', str(node))
        self.register_receipt_callback(rtp_packet_listen, self.ReceiveRTPPacket, (str(node), time_rtp_sent))

# Function called at the final destination after receiving an incoming RTP packet
def ReceiveRTPPacket (self, action, data):
    # Read arguments passed to the function
    receiver, time_rtp_sent = data

    # Check the arrival time
    time_rtp_received = self.sim_time()
    # Calculate packet delay
    delay = time_rtp_received - time_rtp_sent

    # Get the receiver node (terminal/destination) & CSCF node handles
    receiver_node = self.get_tier_node(receiver)

    # Store the voice packet delay as a global statistic
    stat_handle = self.rsap_node.stat_register('REC: RTP voice packet delay (s)', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
    stat_handle.write (delay)

    # Store the voice packet delay as a global statistic
    stat_handle = receiver_node.stat_register('REC: RTP voice packet delay (s)', Aps.Stat_Type.Local, Aps.Stat_Collect_Mode.All_Values)
    stat_handle.write (delay)

    # Jitter calculations
    PreviousDelay = self.rsap_node.get_state ('RECPreviousDelay')
    PreviousJitter = self.rsap_node.get_state ('RECJitter')
```

```
    if PreviousDelay == None:
        # It is the first sample, initialize the arguments
        jitter = 0.0
    else:
        d = abs(delay - PreviousDelay)
        jitter = float(PreviousJitter + ((d - PreviousJitter) / 16))
        stat_handle = receiver_node.stat_register('REC: RTP jitter')
        stat_handle.write (jitter)
        stat_handle = self.rsap_node.stat_register('REC: RTP jitter', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
        stat_handle.write (jitter)

    self.rsap_node.set_state ('RECPreviousDelay', delay)
    self.rsap_node.set_state ('RECJitter', jitter)

    # Local statistic: Number of received RTP packets by the receiver node
    RECPacketsReceived = receiver_node.get_state ('RECPacketsReceived')
    #If there is no RECPacketsReceived state yet. Thus, initiate it.
    if RECPacketsReceived == None:
        RECPacketsReceived = 0
    # RTP packet has been received. Increase the counter.
    RECPacketsReceived += 1
    # Store the counter as a node state for future use
    receiver_node.set_state ('RECPacketsReceived', RECPacketsReceived)
    # Store the counter value as a local statistic
    stat_handle = receiver_node.stat_register('REC: RTP voice packets received')
    stat_handle.write(RECPacketsReceived)

    if (receiver == 'Terminal') or (receiver == 'ControlCenter'):
        # Global: Number of received RTP packets (via RSAP)
        RECTotalPacketsReceived = self.rsap_node.get_state ('RECTotalPacketsReceived')
        #If there is no OnePacketsReceived state yet. Thus, initiate it.
        if RECTotalPacketsReceived == None:
            RECTotalPacketsReceived = 0
        # RTP packet has been received. Increase the counter.
        RECTotalPacketsReceived += 1
        # Store the counter as a node state for future use
        self.rsap_node.set_state ('RECTotalPacketsReceived', RECTotalPacketsReceived)
        # Store the counter value as a local statistic
        stat_handle = self.rsap_node.stat_register('REC: Total RTP voice packets received', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
        stat_handle.write (RECTotalPacketsReceived)

# Function calculating the global performance measures
def CalculateLoss (self, action, data):

    # Read the states (number of received/send packets)
    RECTotalPacketsSent = float(self.rsap_node.get_state ('RECTotalPacketsSent'))
    RECTotalPacketsReceived = float(self.rsap_node.get_state ('RECTotalPacketsReceived'))

    # Calculate the packet loss
    RTPpacketLoss = float((RECTotalPacketsSent - RECTotalPacketsReceived) / RECTotalPacketsSent)
    # Store the packet loss as a statistic
    stat_handle = self.rsap_node.stat_register('REC: RTP voice packet loss', Aps.Stat_Type.Global, Aps.Stat_Collect_Mode.All_Values)
```

```
    stat_handle.write (RTPpacketLoss)

    for node in self.TierList:
        receiver_node = self.get_tier_node(node)
        PacketsReceived = float(receiver_node.get_state ('RECPacketsReceived'))
        RTPpacketLoss = float((RECTotalPacketsSent - PacketsReceived) /
             RECTotalPacketsSent)
        stat_handle = receiver_node.stat_register('REC: RTP voice packet loss')
        stat_handle.write (RTPpacketLoss)

def Ignore (self, action, data):
    pass
```

APPENDIX B

Simulation details

This appendix includes attribute configurations and other details concerning OPNET simulations described in this thesis. Content of this appendix is divided into three sections:

- **Section B.1** includes simulation details that are common for both of the scenarios presented in the thesis.
- **Section B.2** includes details specific to the Snoghøj-Odense scenario.
- **Section B.3** includes simulation details specific to the Copenhagen Central Station scenario.

B.1 Details common for both scenarios

Table B.1: LTE network configuration used in the simulations. These parameters define radio bands used on the LTE radio interface and EPS bearers for the user plane.

lte_attr_definer_adv

Full Name: Funenbanen.LTE network configuration

name	LTE network configuration
EPS Bearer Definitions	(...)
Efficiency Attributes	Physical Layer Enabled
FDD Profiles	(...)
MBSFN Area Profiles	Default
Multipath Channel Definitions	Default (Manual Configuration)
TDD Profiles	Default TDD
eNodeB Failure/Recovery Modeling	Disabled
eNodeB Failure/Recovery Specification	(...)

Attribute: EPS Bearer Definitions

Index	Name	QoS Class Identifier	Allocation Retention Priority	Uplink Guaranteed Bit Rate	Downlink Guaranteed Bit Rate	Uplink Maximum Bit Rate	Downlink Maximum Bit Rate
0	REC Signalling	1 (GBR)	1	64 Kbps	64 Kbps	5 Mbps	5 Mbps
1	REC Media	1 (GBR)	1	64 Kbps	64 Kbps	5 Mbps	5 Mbps
2	ETCS bearer	3 (GBR)	3	32 Kbps	32 Kbps	1 Mbps	1 Mbps
3	One-to-One Signalling	2 (GBR)	4	64 Kbps	64 Kbps	5 Mbps	5 Mbps
4	One-to-One Media	2 (GBR)	5	64 Kbps	64 Kbps	5 Mbps	5 Mbps

Attribute: FDD Profiles

Index	Name	UL SC-FDMA Channel Configuration	DL OFDMA Channel Configuration
0	LTE 20 MHz FDD	Default UL 20 Mhz	Default DL 20 MHz
1	LTE 15 MHz FDD	Default UL 15 Mhz	Default DL 15 MHz
2	LTE 10 MHz FDD	Default UL 10 Mhz	Default DL 10 MHz
3	LTE 5 MHz FDD	Default UL 5 Mhz	Default DL 5 MHz
4	LTE 3 MHz FDD	Default UL 3 Mhz	Default DL 3 MHz
5	LTE 1.4 MHz FDD	Default UL 1.4 Mhz	Default DL 1.4 MHz
6	LTE 5 MHz FDD 900 MHz	(...)	(...)
7	LTE 5 MHz FDD 5.9 GHz A	(...)	(...)
8	LTE 5 MHz FDD 5.9 GHz B	(...)	(...)

Attribute: FDD Profiles.UL SC-FDMA Channel Configuration [6]

Base Frequency	0.878
Bandwidth	5 MHz
Cyclic Prefix Type	Normal (7 Symbols per Slot)

Attribute: FDD Profiles.DL OFDMA Channel Configuration [6]

Base Frequency	0.923
Bandwidth	5 MHz
Cyclic Prefix Type	Normal (7 Symbols per Slot)

Attribute: FDD Profiles.UL SC-FDMA Channel Configuration [7]

Base Frequency	5.9075
Bandwidth	5 MHz
Cyclic Prefix Type	Normal (7 Symbols per Slot)

Attribute: FDD Profiles.DL OFDMA Channel Configuration [7]

Base Frequency	5.9225
Bandwidth	5 MHz
Cyclic Prefix Type	Normal (7 Symbols per Slot)

Attribute: FDD Profiles.UL SC-FDMA Channel Configuration [8]

Base Frequency	5.9175
Bandwidth	5 MHz
Cyclic Prefix Type	Normal (7 Symbols per Slot)

Attribute: FDD Profiles.DL OFDMA Channel Configuration [8]

Base Frequency	5.9325
Bandwidth	5 MHz
Cyclic Prefix Type	Normal (7 Symbols per Slot)

Table B.2: UE node (train node) attribute configuration. The same configuration was used in both scenarios with the exception of movement trajectory (used only in Snoghøj-Odense).

lte_wkstn_adv

Full Name: Funenbanen.Train 1

name	Train 1
trajectory	Snoghoj-Odense L1 180
AODV Parameters	Default
Antenna Gain (dBi)	0.0
Application: Destination Preferences	(...)
Application: Multicasting Specification	None
Application: RSVP Parameters	None
Application: Segment Size	64,000
Application: Source Preferences	None
Application: Supported Profiles	(...)
Application: Supported Services	None
Application: Transaction Model Tier Configuration	(...)
Application: Transport Protocol Specification	Default
Assigned Administrative Domain	opnet.com
Assigned Gatekeeper	Auto Assigned
Battery Capacity	Unlimited
CPU Background Utilization	None
CPU Resource Parameters	(...)
Call Signaling Mode	Direct Endpoint Call Signaling
Client Address	Auto Assigned
DRX Parameters	DRX Enabled with Serving Cell Parameters
DSR Parameters	Default
DVMRP Parameters	Not Configured
EPS Bearer Configurations	(...)
H323 Device Role	Terminal
HARQ Parameters	(...)
Handover Parameters	Same as Serving eNodeB
IGMP Parameters	Default
IMSI	Auto Assigned
IP Forwarding Table	Do Not Export
IP Host Parameters	(...)
IP Processing Information	(...)
IP QoS Parameters	None
Link Adaptation Parameters	(...)
Link Monitoring Parameters	Default
Max Number of Calls	Unlimited
Maximum Transmission Power (W)	2.0
Minimum Available Bandwidth	LTE
Mobile IPv4 Parameters	Disabled
Mobile IPv6 Parameters	Not Configured
Modulation and Coding Scheme Index	20
Multipath Channel Model (Downlink)	LTE OFDMA ITU Vehicular A
Multipath Channel Model (Uplink)	LTE SCFDMA ITU Vehicular A
Number of Receive Antennas	1
Number of Transmit Antennas	1
OLSR Parameters	Default
Operational Power Settings	Default
PDCP Compression	Disabled
Pathloss Parameters	(...)
RSVP Protocol Parameters	(...)
Receiver Sensitivity (dBm)	-130dBm

Reporting End Time	Use Global Setting
Reporting Start Time	Use Global Setting
SIP UAC Parameters	(...)
Server: Advanced Server Configuration	Sun Ultra 10 333MHz:: 1 CPU, 1 Core(s) Per CPU, 333MHz, Solaris, System
Server: Modeling Method	Simple CPU
Serving EPC ID	0
Serving eNodeB ID	Perform Cell Search
TCP Parameters	(...)
TORA/IMEP Parameters	Default
Timers	Default
VRF Table	Do Not Export
eNodeB Selection Policy	Best Suitable eNodeB

Attribute: Application: Destination Preferences

Index	Application	Symbolic Name	Actual Name
0	CCTV	Video Destination	(...)
1	Voice announcements	Voice Destination	(...)
2	File update	FTP Server	(...)
3	ETCSApp	RBC	(...)
4	VoLTE Call	CSCF	(...)
5	REC Call	CSCF	(...)

Attribute: Application: Destination Preferences.Actual Name [0]

Name	Funenbanen.Server_CCTV
Selection Weight	10

Attribute: Application: Destination Preferences.Actual Name [1]

Name	Funenbanen.Server_V_Announcements
Selection Weight	10

Attribute: Application: Destination Preferences.Actual Name [2]

Name	Funenbanen.Server_File_Transfer_Real_Time_Info
Selection Weight	10

Attribute: Application: Destination Preferences.Actual Name [3]

Name	Funenbanen.RBC
Selection Weight	10

Attribute: Application: Destination Preferences.Actual Name [4]

Name	Funenbanen.CSCF
Selection Weight	10

Attribute: Application: Destination Preferences.Actual Name [5]

Name	Funenbanen.CSCF
Selection Weight	10

Attribute: Application: Supported Profiles

Profile Name	RollingStockWithREC
Traffic Type	All Discrete

Application Delay Tracking	Disabled

Attribute: Application: Transaction Model Tier Configuration

Index	Application	Tier Name	Popularity	Advanced Server Model Integration
0	ETCSApp	OBU	10	Not Used
1	VoLTE Call	Terminal	10	Not Used
2	VoLTE Call	Destination	10	Not Used
3	REC Call	Terminal	10	Not Used

Attribute: CPU Resource Parameters

Number of Resources	1
Task Contention Mode	Contention Already Modeled
Processing Speed Multiplier	1.0
Multi-tasking Performance Table	No Entries

Attribute: EPS Bearer Configurations

Index	Bearer Name	TFT Packet Filters	Radio Bearer RLC Configuration	Action If Not Admitted	Radio Bearer PDCP Configuration
0	REC Signalling	(...)	(...)	Discard Data	Default
1	REC Media	(...)	(...)	Discard Data	Default
2	ETCS bearer	(...)	(...)	Discard Data	Default
3	VoLTE Signalling	(...)	(...)	Discard Data	Default
4	VoLTE Media	(...)	(...)	Discard Data	Default

Attribute: EPS Bearer Configurations.TFT Packet Filters [0]

Index	Match Property	Match Value	Direction
0	IP ToS	Interactive Voice (6)	Bidirectional
1	Protocol	TCP	Bidirectional

Attribute: EPS Bearer Configurations.Radio Bearer RLC Configuration [0]

Index	Direction	Mode	Reordering Timer Duration	Poll Retransmit Timer Duration	Status Report Prohibit Timer Duration	Maximum PDUs without Poll	Maximum Bytes without Poll	Maximum Number of Retransmissions
0	Uplink	Acknowledged	35	45	0	Infinity	Infinity	3
1	Downlink	Acknowledged	35	45	0	Infinity	Infinity	3

Attribute: EPS Bearer Configurations.TFT Packet Filters [1]

Index	Match Property	Match Value	Direction
0	IP ToS	Interactive Voice (6)	Bidirectional
1	Protocol	UDP	Bidirectional

Attribute: EPS Bearer Configurations.Radio Bearer RLC Configuration [1]

Index	Direction	Mode	Reordering Timer Duration	Poll Retransmit Timer Duration	Status Report Prohibit Timer Duration	Maximum PDUs without Poll	Maximum Bytes without Poll	Maximum Number of Retransmissions
0	Uplink	Acknowledged	35	45	0	Infinity	Infinity	3
1	Downlink	Acknowledged	35	45	0	Infinity	Infinity	3

Attribute: EPS Bearer Configurations.TFT Packet Filters [2]

Match Property	IP ToS
Match Value	Excellent Effort (3)
Direction	Bidirectional

Attribute: EPS Bearer Configurations.Radio Bearer RLC Configuration [2]

Index	Direction	Mode	Reordering Timer Duration	Poll Retransmit Timer Duration	Status Report Prohibit Timer Duration	Maximum PDUs without Poll	Maximum Bytes without Poll	Maximum Number of Retransmissions
0	Uplink	Acknowledged	35	45	0	Infinity	Infinity	3
1	Downlink	Acknowledged	35	45	0	Infinity	Infinity	3

Attribute: EPS Bearer Configurations.TFT Packet Filters [3]

Index	Match Property	Match Value	Direction
0	IP ToS	Interactive Multimedia (5)	Bidirectional
1	Protocol	TCP	Bidirectional

Attribute: EPS Bearer Configurations.Radio Bearer RLC Configuration [3]

Index	Direction	Mode	Reordering Timer Duration	Poll Retransmit Timer Duration	Status Report Prohibit Timer Duration	Maximum PDUs without Poll	Maximum Bytes without Poll	Maximum Number of Retransmissions
0	Uplink	Acknowledged	35	45	0	Infinity	Infinity	3
1	Downlink	Acknowledged	35	45	0	Infinity	Infinity	3

Attribute: EPS Bearer Configurations.TFT Packet Filters [4]

Index	Match Property	Match Value	Direction
0	IP ToS	Interactive Multimedia (5)	Bidirectional
1	Protocol	UDP	Bidirectional

Attribute: EPS Bearer Configurations.Radio Bearer RLC Configuration [4]

Index	Direction	Mode	Reordering Timer Duration	Poll Retransmit Timer Duration	Status Report Prohibit Timer Duration	Maximum PDUs without Poll	Maximum Bytes without Poll	Maximum Number of Retransmissions
0	Uplink	Unacknowledged	35	45	0	Infinity	Infinity	4
1	Downlink	Unacknowledged	35	45	0	Infinity	Infinity	4

Attribute: HARQ Parameters

Uplink Parameters	(...)
Downlink Parameters	(...)

Attribute: HARQ Parameters.Uplink Parameters

Max Retransmissions	4

Attribute: HARQ Parameters.Downlink Parameters

Max Retransmissions	4

Attribute: IP Host Parameters

Interface Information	(...)
Passive RIP Routing	Disabled
Default Route	Auto Assigned
Static Routing Table	None
IPv6 Default Route	Auto Assigned
Multicast Mode	Disabled
IPv6 Static Routing Table	(...)

Attribute: IP Host Parameters.Interface Information

Name	IF0
Address	Auto Assigned
Subnet Mask	Auto Assigned
MTU	1500
Compression Information	None
IPv6 Parameters	(...)
Description	N/A
Layer 2 Mappings	None

Attribute: IP Host Parameters.Interface Information.IPv6 Parameters

Link-Local Address	Not Active
Global Address(es)	None
Router Solicitation Parameters	Default
Neighbor Cache Parameters	Default
MTU	Ethernet

Attribute: IP Processing Information

Processing Scheme	Central Processing
Backplane Transfer Rate	Not Used
Datagram Switching Rate	500,000
Datagram Forwarding Rate	Infinity
Forwarding Rate Units	packets/second
Memory Size	16 MB

Attribute: Link Adaptation Parameters

Measurement Window Size	100 ms
Downlink Target Link Quality Measure	1%

Attribute: Pathloss Parameters

Pathloss Model	Suburban Macrocell (3GPP)
Model Arguments	Not Applicable
Shadow Fading	Suburban Macrocell (3GPP) Default

Attribute: TCP Parameters

Host Operating System	Windows 7
Flavor	New Reno
Maximum Segment Size	Auto-Assigned
Receive Buffer	Auto-Tuning
Receive Buffer Adjustment	Windows Based
Receive Buffer Usage Threshold	0.0

Delayed ACK Mechanism	Segment/Clock Based
Maximum ACK Delay	0.200
Maximum ACK Segments	2
Slow-Start Initial Count	2
Duplicate ACK Threshold	3
Window Scaling	Enabled
Selective ACK (SACK)	Enabled
Duplicate SACK (D-SACK)	Enabled
ECN Capability	Disabled
Segment Send Threshold	MSS Boundary
Active Connection Threshold	Unlimited
Nagle Algorithm	Enabled
Karn's Algorithm	Enabled
Timestamp	(...)
Initial Sequence Number	Auto Compute
Retransmission Thresholds	(...)
Initial RTO	0.3
Minimum RTO	0.2
Maximum RTO	1.0
RTT Gain	0.125
Deviation Gain	0.25
RTT Deviation Coefficient	4.0
Timer Granularity	0.2
Persistence Timeout	1.0
Connection Information	Do Not Print
Acceleration	Disabled

Attribute: TCP Parameters.Timestamp

Status	Passive
Clock Tick	500

Attribute: TCP Parameters.Retransmission Thresholds

Mode	Attempts Based
Maximum Connect Attempts	7
Maximum Data Attempts	5
Maximum Connect Interval	Not Applicable
Maximum Data Interval	Not Applicable

Table B.3: Ethernet links used in the backbone wired network. The high capacity of these links ensured that potential communication bottlenecks appear in the radio network in the first place.

100Gbps_Ethernet

Full Name: Funenbanen.node_0 <-> EPC_Fredericia

name	node_0 <-> EPC_Fredericia
model	100Gbps_Ethernet
transmitter a	node_0.eth_tx_00_0
receiver a	node_0.eth_rx_00_0
transmitter b	EPC_Fredericia.eth_tx_10_0
receiver b	EPC_Fredericia.eth_rx_10_0
Traffic Information	None

Table B.4: Tasks configuration, which defines the parameters used by the implemented applications models: ETCS signalling, One-to-one VoLTE call and Railway Emergency Call (REC). Details on these models are presented in Appendix A on page 151.

Task Config

Full Name: Funenbanen.Task Configuration

name	Task Configuration
Task Specification	(...)

Attribute: Task Specification

Index	Task Name	Manual Configuration	Connection Policy	Transaction Model Name	Transaction Model Parameters
0	ETCStask	Not Applicable	Reuse Across Tasks	ETCSsignalling.aed.m	MA_size:128;MA_number:20;MA_interval:30.000000;RetransmissionTimeout:0.500000;MaxRetransmissions:7
1	VoLTEtask	Not Applicable	Reuse Across Tasks	One2One.aed.m	VoiceFrameSize:44;CallDuration:20;VoiceFrames:50
2	RECtask	Not Applicable	Reuse Across Tasks	REC.aed.m	VoiceFrameSize:44;CallDuration:60;VoiceFrames:50

Table B.5: Application configuration used in the simulations. These parameters define traffic generated by each of the applications.

Application Config

Full Name: Funenbanen.RailwayApplicationsConfig

name	RailwayApplicationsConfig
Application Definitions	(...)
MOS Advantage Factors	Default
Voice Conversation Environments	All Environments
Voice Encoder Schemes	All Schemes

Attribute: Application Definitions

Index	Name	Description
0	ETCS Signalling	(...)
1	VoLTE Call	(...)
2	Tele maintenance	(...)
3	Passenger information	(...)
4	Video surveillance	(...)

Attribute: Application Definitions.Description [0]

Custom	(...)
Database	Off
Email	Off
Ftp	Off
Http	Off
Print	Off
Peer-to-peer File Sharing	Off
Remote Login	Off
Video Conferencing	Off
Voice	Off

Attribute: Application Definitions.Description [0].Custom

Task Description	(...)
Task Ordering	Serial (Ordered)
Transport Protocol	TCP
Transport Port	Default
Type of Service	Excellent Effort (3)
Connection Policy	Refresh After Application
RSVP Parameters	None
Transaction Model Task Interdependency	Trace Playback Completion for Current Task

Attribute: Application Definitions.Description [0].Custom.Task Description

Task Name	ETCStask
Task Weight	10

Attribute: Application Definitions.Description [1]

Custom	(...)
Database	Off
Email	Off
Ftp	Off
Http	Off
Print	Off

Peer-to-peer File Sharing	Off
Remote Login	Off
Video Conferencing	Off
Voice	Off

Attribute: Application Definitions.Description [1].Custom

Task Description	(...)
Task Ordering	Serial (Ordered)
Transport Protocol	TCP
Transport Port	Default
Type of Service	Interactive Multimedia (5)
Connection Policy	Refresh After Application
RSVP Parameters	None
Transaction Model Task Interdependency	Final Response Arrival at Client

Attribute: Application Definitions.Description [1].Custom.Task Description

Task Name	VoLTEtask
Task Weight	10

Attribute: Application Definitions.Description [2]

Custom	Off
Database	Off
Email	Off
Ftp	(...)
Http	Off
Print	Off
Peer-to-peer File Sharing	Off
Remote Login	Off
Video Conferencing	Off
Voice	Off

Attribute: Application Definitions.Description [2].Ftp

Command Mix	50%
Inter-Request Time	constant (450)
File Size	constant (1024)
Symbolic Server Name	FTP Server
Type of Service	Best Effort (0)
RSVP Parameters	None
Back-End Custom Application	Not Used

Attribute: Application Definitions.Description [3]

Custom	Off
Database	Off
Email	Off
Ftp	Off
Http	Off
Print	Off
Peer-to-peer File Sharing	Off
Remote Login	Off
Video Conferencing	Off
Voice	(...)

Attribute: Application Definitions.Description [3].Voice

Silence Length	(...)
Talk Spurt Length	(...)
Symbolic Destination Name	Voice Destination
Encoder Scheme	GSM FR
Voice Frames per Packet	1
Type of Service	Best Effort (0)
RSVP Parameters	None
Traffic Mix	All Discrete
Signaling	None
Compression Delay	0.02
Decompression Delay	0.02
Conversation Environment	(...)

Attribute: Application Definitions.Description [3].Voice.Silence Length

Incoming Silence Length	exponential (0.65)
Outgoing Silence Length	constant (0.99)

Attribute: Application Definitions.Description [3].Voice.Talk Spurt Length

Incoming Talk Spurt Length	exponential (0.352)
Outgoing Talk Spurt Length	constant (0.01)

Attribute: Application Definitions.Description [3].Voice.Conversation Environment

Incoming Conversation Environment	Land phone - Quiet room
Outgoing Conversation Environment	Land phone - Quiet room

Attribute: Application Definitions.Description [4]

Custom	Off
Database	Off
Email	Off
Ftp	Off
Http	Off
Print	Off
Peer-to-peer File Sharing	Off
Remote Login	Off
Video Conferencing	(...)
Voice	Off

Attribute: Application Definitions.Description [4].Video Conferencing

Frame Interarrival Time Information	(...)
Frame Size Information	(...)
Symbolic Destination Name	Video Destination
Type of Service	Best Effort (0)
RSVP Parameters	None
Traffic Mix	All Discrete

Attribute: Application Definitions.Description [4].Video Conferencing.Frame Interarrival Time Information

Incoming Stream Interarrival Time	constant (0.05)
Outgoing Stream Interarrival Time	constant (0.05)

Attribute: Application Definitions.Description [4].Video Conferencing.Frame Size Information

Incoming Stream Frame Size	constant (3125)
Outgoing Stream Frame Size	constant (3125)

Table B.6: User application profiles, which specify which applications are used by which user type. Moreover, these profiles define when and how often each application is executed.

Profile Config

Full Name: Funenbanen.RailwayUserProfiles

name	RailwayUserProfiles
Profile Configuration	(...)

Attribute: Profile Configuration

Index	Profile Name	Applications	Operation Mode	Start Time	Duration	Repeatability
0	RollingStock	(...)	Simultaneous	uniform (100, 110)	End of Simulation	Once at Start Time
1	RollingStockWithREC	(...)	Simultaneous	uniform (100, 110)	End of Simulation	Once at Start Time

Attribute: Profile Configuration.Applications [0]

Index	Name	Start Time Offset	Duration	Repeatability
0	Video surveillance	uniform (30, 60)	End of Profile	Once at Start Time
1	Passenger information	exponential (450)	constant (5)	(...)
2	Tele maintenance	exponential (60)	End of Last Task	Once at Start Time
3	ETCS Signalling	uniform (20, 30)	End of Last Task	Once at Start Time
4	VoLTE Call	exponential (400)	End of Last Task	(...)

Attribute: Profile Configuration.Applications [0].Repeatability [1]

Inter-repetition Time	exponential (900)
Number of Repetitions	Unlimited
Repetition Pattern	Serial

Attribute: Profile Configuration.Applications [0].Repeatability [4]

Inter-repetition Time	exponential (600)
Number of Repetitions	Unlimited
Repetition Pattern	Serial

Attribute: Profile Configuration.Applications [1]

Index	Name	Start Time Offset	Duration	Repeatability
0	Video surveillance	uniform (30, 60)	End of Profile	Once at Start Time
1	Passenger information	exponential (450)	constant (5)	(...)
2	Tele maintenance	exponential (60)	End of Last Task	Once at Start Time
3	ETCS Signalling	uniform (20, 30)	End of Last Task	Once at Start Time
4	VoLTE Call	exponential (400)	End of Last Task	(...)
5	REC Call	uniform (60, 400)	End of Profile	Once at Start Time

Attribute: Profile Configuration.Applications [1].Repeatability [1]

Inter-repetition Time	exponential (900)
Number of Repetitions	Unlimited
Repetition Pattern	Serial

Attribute: Profile Configuration.Applications [1].Repeatability [4]

Inter-repetition Time	exponential (600)
Number of Repetitions	Unlimited
Repetition Pattern	Serial

B.2 Details of the Snoghøj-Odense scenario

Figure B.1: Current deployment of GSM-R base stations along Snoghøj-Odense line. Base stations are marked as the dark grey dots.

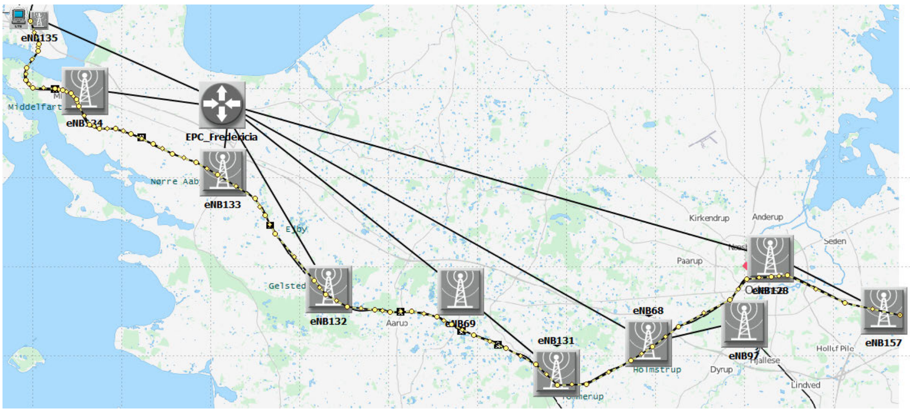


Figure B.2: UE (train) trajectory illustrated by the yellow dotted line on top of the Snoghøj-Odense map

Table B.7: Detailed trajectory file specifying the UE (train) movement in Snoghøj-Odense simulation scenario. Half of the UEs (trains) were following the trajectory from the West to the East, while the other half in the opposite direction. After completing the whole trajectory, an UE was following it again in the opposite direction. UEs (trains) travelling in one direction were following each other with a minimum headway of 3 min.

	X Pos (km)	Y Pos (km)	Distance (km)	Altitude (m)	Traverse Time	Ground Speed	Ascent Rate (m/sec)	Wait Time	Accum Time	Pitch (degrees)	Yaw (degrees)	Roll (degrees)
1	-20.146026	18.729976	n/a	4.000	n/a	n/a	n/a	3m00.00s	3m00.00s	Autocomputed	Autocomputed	Unspecified
2	-19.664165	17.457371	1.3628	4.000	27.26s	179.9780	0	0	3m27.26s	Autocomputed	Autocomputed	Unspecified
3	-19.701232	17.086709	0.3723	4.000	07.45s	179.9268	0	0	3m34.71s	Autocomputed	Autocomputed	Unspecified
4	-20.343712	16.098278	1.1764	4.000	23.53s	179.9901	0	0	3m58.24s	Autocomputed	Autocomputed	Unspecified
5	-20.417844	15.418731	0.6832	4.000	13.66s	180.0641	0	0	4m11.90s	Autocomputed	Autocomputed	Unspecified
6	-20.010116	14.998648	0.5868	4.000	11.74s	179.9281	0	0	4m23.64s	Autocomputed	Autocomputed	Unspecified
7	-18.749866	14.949227	1.2614	4.000	25.23s	179.9910	0	0	4m48.87s	Autocomputed	Autocomputed	Unspecified
8	-18.317428	14.875094	0.4391	4.000	08.78s	180.0227	0	0	4m57.65s	Autocomputed	Autocomputed	Unspecified
9	-17.860278	14.702119	0.4894	4.000	09.79s	179.9805	0	0	5m07.44s	Autocomputed	Autocomputed	Unspecified
10	-17.551393	14.368523	0.4556	4.000	09.11s	180.0211	0	0	5m16.55s	Autocomputed	Autocomputed	Unspecified
11	-17.427840	13.960795	0.4265	4.000	08.53s	180.0047	0	0	5m25.08s	Autocomputed	Autocomputed	Unspecified
12	-16.810070	12.910587	1.2205	4.000	24.41s	180.0019	0	0	5m49.49s	Autocomputed	Autocomputed	Unspecified
13	-16.278788	12.700545	0.5720	4.000	11.44s	180.0097	0	0	6m00.93s	Autocomputed	Autocomputed	Unspecified
14	-15.352134	12.712901	0.9267	4.000	18.53s	180.0374	0	0	6m19.46s	Autocomputed	Autocomputed	Unspecified
15	-14.499612	12.428727	0.8996	4.000	17.99s	180.0107	0	0	6m37.45s	Autocomputed	Autocomputed	Unspecified
16	-13.239362	11.835668	1.3945	4.000	27.89s	180.0019	0	0	7m05.34s	Autocomputed	Autocomputed	Unspecified
17	-12.152087	11.292031	1.2170	4.000	24.34s	180.0022	0	0	7m29.68s	Autocomputed	Autocomputed	Unspecified
18	-10.817705	10.822526	1.4157	4.000	28.31s	180.0290	0	0	7m57.99s	Autocomputed	Autocomputed	Unspecified
19	-9.606877	10.142979	1.3899	4.000	27.80s	179.9821	0	0	8m25.79s	Autocomputed	Autocomputed	Unspecified
20	-7.506460	8.771531	2.5107	4.000	50.21s	180.0176	0	0	9m16.00s	Autocomputed	Autocomputed	Unspecified
21	-6.938112	7.844876	1.0879	4.000	21.76s	179.9769	0	0	9m37.76s	Autocomputed	Autocomputed	Unspecified
22	-6.369764	6.819379	1.1732	4.000	23.46s	180.0328	0	0	10m01.22s	Autocomputed	Autocomputed	Unspecified
23	-4.207570	4.187680	3.4080	4.000	1m08.16s	180.0010	0	0	11m09.38s	Autocomputed	Autocomputed	Unspecified
24	-3.552735	3.396935	1.0271	4.000	20.54s	180.0231	0	0	11m29.92s	Autocomputed	Autocomputed	Unspecified
25	-2.317195	2.729744	1.4046	4.000	28.09s	180.0082	0	0	11m58.01s	Autocomputed	Autocomputed	Unspecified
26	-0.797482	2.544413	1.5310	4.000	30.62s	180.0044	0	0	12m28.63s	Autocomputed	Autocomputed	Unspecified
27	0.067395	2.420859	0.8737	4.000	17.47s	180.0347	0	0	12m46.10s	Autocomputed	Autocomputed	Unspecified
28	1.278223	2.433215	1.2109	4.000	24.22s	179.9841	0	0	13m10.32s	Autocomputed	Autocomputed	Unspecified
29	2.686738	2.161396	1.4344	4.000	28.69s	179.9853	0	0	13m39.01s	Autocomputed	Autocomputed	Unspecified
30	3.539260	1.716602	0.9613	4.000	19.23s	179.9628	0	0	13m58.24s	Autocomputed	Autocomputed	Unspecified
31	4.762444	1.135899	1.3535	4.000	27.07s	180.0047	0	0	14m25.31s	Autocomputed	Autocomputed	Unspecified
32	6.591042	0.369864	1.9817	4.000	39.63s	180.0145	0	0	15m04.94s	Autocomputed	Autocomputed	Unspecified
33	8.246664	-0.507368	1.8724	4.000	37.45s	179.9867	0	0	15m42.39s	Autocomputed	Autocomputed	Unspecified
34	9.457493	-1.681131	1.6847	4.000	33.69s	180.0186	0	0	16m16.08s	Autocomputed	Autocomputed	Unspecified
35	11.100760	-1.631709	1.6441	4.000	32.88s	180.0137	0	0	16m48.96s	Autocomputed	Autocomputed	Unspecified
36	13.596549	-0.322038	2.8218	4.000	56.44s	179.9868	0	0	17m45.40s	Autocomputed	Autocomputed	Unspecified
37	14.832088	0.481063	1.4758	4.000	29.52s	179.9733	0	0	18m14.92s	Autocomputed	Autocomputed	Unspecified
38	16.265314	1.593048	1.8171	4.000	36.34s	180.0112	0	0	18m51.26s	Autocomputed	Autocomputed	Unspecified
39	19.168831	3.112761	3.2826	4.000	1m05.65s	180.0054	0	0	19m56.91s	Autocomputed	Autocomputed	Unspecified
40	20.120196	4.249457	1.4855	4.000	29.71s	180.0054	0	0	20m26.62s	Autocomputed	Autocomputed	Unspecified
41	21.516355	4.385367	1.4034	4.000	28.07s	179.9869	0	0	20m54.69s	Autocomputed	Autocomputed	Unspecified
42	22.418299	4.484210	0.9078	4.000	18.16s	179.9670	0	0	21m12.85s	Autocomputed	Autocomputed	Unspecified
43	25.643056	2.939786	3.5679	4.000	1m11.36s	179.9959	0	0	22m24.21s	Autocomputed	Autocomputed	Unspecified
44	28.756615	2.223173	3.1907	4.000	1m03.81s	180.0086	0	0	23m28.02s	Autocomputed	Autocomputed	Unspecified

Table B.8: eNodeB node attribute configuration used in Snoghøj-Odense simulation scenario. Maximum transmission power was defined for each deployment case following the relation shown in Figure 3.13 on page 55.

lte_enodeb_ethernet4_adv

Full Name: Funenbanen.eNB135

name	eNB135
AAA Parameters	Not Configured
APS Parameters	None
ARP Parameters (IF0 P0)	Default
ARP Parameters (IF1 P0)	Default
ARP Parameters (IF2 P0)	Default
ARP Parameters (IF3 P0)	Default
Address	Auto Assigned
Admission Control Parameters	Default
Antenna Gain (dBi)	15 dBi
BGP Based	None
BGP Parameters	Default
BGP Routing Table	Do Not Export
Battery Capacity	Unlimited
Buffer Status Report Parameters	Default
CPU Background Utilization	None
CPU Resource Parameters	Single Processor
CPU Utilization	Unassigned
CQI Transmission Parameters	Default
Cross Connect Groups	None
Cross-Connects Parameters	None
Customers	Not Configured
DHCPv6 Client Parameters	Disabled
DHCPv6 Server Parameters	Disabled
DLSw+ Parameters	None
DNS Parameters	None
DRX Parameters	Default
DVMRP Parameters	Not Configured
Delay, Jitter and Loss	Unassigned
EIGRP Parameters	(...)
EIGRP Routing Table	Do Not Export
EPCs Served	All
Ethernet Parameters (IF0 P0)	Default (Host)
Ethernet Parameters (IF1 P0)	Default (Host)
Ethernet Parameters (IF2 P0)	Default (Host)
Ethernet Parameters (IF3 P0)	Default (Host)
HSRP Operational Data	Unknown
HSRP Parameters	Not Configured
Handover Parameters	Default
IGMP Operational Data	None
IGMP Parameters	Not Configured
IGRP Parameters	(...)
IGRP Routing Table	Do Not Export
IP Forwarding Table	Do Not Export
IP Multicast Group-to-RP Table	Do Not Export
IP Multicast Parameters	Not Configured
IP QoS Parameters	None
IP Routing Parameters	(...)
IPSec Parameters	None
IPX Parameters	None
IPv6 Parameters	None
IS-IS Parameters	(...)

IS-IS Routing Table	Do Not Export
Kerberos Parameters	Not Configured
L1/L2 Control Parameters	Default
L2TP Control Channel Parameters	None
LACP System Priority	32768
LDP Based	None
LDP Parameters	(...)
Line Information	None
Logging	Disabled
MBSFN Area	Disabled
MPLS Parameters	(...)
MSDP Parameters	Not Configured
Maximum Transmission Power (W)	promoted
NAT Parameters	Not Configured
NHRP Parameters	None
Number of Receive Antennas	2
Number of Transmit Antennas	2
OSPF Link State Database	Do Not Export
OSPF Parameters	(...)
OSPF Routing Table	Do Not Export
OSPFv3 Parameters	(...)
Operating Power	20
Operational HTTP Parameters	Not Configured
PDCP Compression	Disabled
PHY Profile	LTE 5 MHz FDD 900 MHz
PIM Parameters	Not Configured
PIM-DVMRP Interoperability Parameters	Not Configured
PIM-SM Routing Table	Do Not Export
Pathloss Parameters	(...)
Pseudowire Classes	None
RADIUS Parameters	Not Configured
RAM Utilization	Unassigned
RIP Parameters	(...)
RIP Routing Table	Do Not Export
RIPng Parameters	(...)
RIPng Routing Table	Do Not Export
RSRB Parameters	None
RSVP Protocol Parameters	(...)
Random Access Parameters	Default
Receiver Sensitivity (dBm)	-200dBm
SNMP Parameters	Disabled
SRP Parameters	Not Configured
Scheduling Mode	Link Adaptation and Channel Dependent Scheduling
Service Distribution Points	Not Configured
Serving MBMS EPC ID	0
Subscriber Services	Not Configured
System Information	(...)
TACACS+ Parameters	Not Configured
TCP Parameters	(...)
Transparent Bridge Parameters	None
User Access Control	Not Configured
VRF Instances	None
VRF Table	Do Not Export
VRRP Parameters	Not Configured
WLAN Configuration	Not Configured
X2 Capability	Enabled

eNodeB ID	1350
eNodeB Selection Threshold	-110dBm
lte_as.Sector Number	promoted
lte_s1.Sector Number	promoted
sysmgt.Device Availability	Unassigned
sysmgt.Fabric Extenders	promoted
sysmgt.Interface Availability	Unnasigned
sysmgt.Packet Errors and Discards	(...)
sysmgt.Throughput	Unassigned
sysmgt.Traffic Statistics	Unassigned
sysmgt.VDC Configuration	Not Configured

Attribute: IP Routing Parameters

Router ID	0.0.0.1
Autonomous System Number	Auto Assigned
Interface Information	(...)
Aggregate Interfaces	None
Loopback Interfaces	(...)
Tunnel Interfaces	None
VLAN Interfaces	None
BVI Interfaces	None
Service Interfaces	None
IPSec Connections	None
Controller Configuration	None
Default Gateway	Unassigned
Default Network(s)	None
Static Routing Table	None
Static Routes Across VRFs	Enabled
Load Balancing Options	Destination-Based
Multipath Routes Threshold	Unlimited
Administrative Weights	(....)
OS Version	Not Set
Standard ACL Configuration	None
Extended ACL Configuration	None
Damping Configuration	None
AS Path Lists	None
Community Lists	None
Extended Community Lists	None
Prefix Filter Configuration	None
Route Map Configuration	None
Route Policy Configuration	None
Firewall Filter Configuration	None
Local Policy	None
Forwarding Table Policies	Not Configured
Fate Sharing	Not Configured
IPv4 Configurations	Default
IP Domain Lookup	Not Configured
Tunnel Policy Configuration	Not Configured

Attribute: IP Routing Parameters.Loopback Interfaces

Name	LB0
Status	Active
Operational Status	Infer
Address	Auto Assigned
Subnet Mask	Auto Assigned

Secondary Address Information	Not Used
Routing Protocol(s)	RIP
MTU	Ethernet
MTU Intended	disabled
Protocol MTUs	(...)
Metric Information	Default
Packet Filter	None
Policy Routing	None
Routing Instance	None
VRF Sitemap	None
Interface Configurations	Default
Description	N/A

Attribute: OSPF Parameters

Processes	Default
Interface Information	(...)
Aggregate Interfaces	None
Loopback Interfaces	(...)
Tunnel Interfaces	None
VLAN Interfaces	None
BVI Interfaces	None

Attribute: OSPF Parameters.Interface Information.Timers [0]

Hello Interval	10
Router Dead Interval	40
Interface Transmission Delay	1.0
Retransmission Interval	5.0

Attribute: OSPF Parameters.Interface Information.MANET Parameters [0]

Full Hello Frequency	Every Third
Hello Repeat Count	Thrice
Adjacency Connectivity	Biconnected
MDR Constraint	3

Attribute: OSPF Parameters.Interface Information.Timers [1]

Hello Interval	10
Router Dead Interval	40
Interface Transmission Delay	1.0
Retransmission Interval	5.0

Attribute: OSPF Parameters.Interface Information.MANET Parameters [1]

Full Hello Frequency	Every Third
Hello Repeat Count	Thrice
Adjacency Connectivity	Biconnected
MDR Constraint	3

Attribute: OSPF Parameters.Interface Information.Timers [2]

Hello Interval	10
Router Dead Interval	40
Interface Transmission Delay	1.0
Retransmission Interval	5.0

Attribute: OSPF Parameters.Interface Information.MANET Parameters [2]

Full Hello Frequency	Every Third
Hello Repeat Count	Thrice
Adjacency Connectivity	Biconnected
MDR Constraint	3

Attribute: OSPF Parameters.Interface Information.Timers [3]

Hello Interval	10
Router Dead Interval	40
Interface Transmission Delay	1.0
Retransmission Interval	5.0

Attribute: OSPF Parameters.Interface Information.MANET Parameters [3]

Full Hello Frequency	Every Third
Hello Repeat Count	Thrice
Adjacency Connectivity	Biconnected
MDR Constraint	3

Attribute: OSPF Parameters.Loopback Interfaces

Name	LB0
Status	Disabled
Silent Mode	Disabled
Area ID	0
Process Tag(s)	1

Attribute: OSPFv3 Parameters

Processes	Default
Interface Information	(...)
Aggregate Interfaces	None
Loopback Interfaces	None
Tunnel Interfaces	None
VLAN Interfaces	None

Attribute: Pathloss Parameters

Pathloss Model	Suburban Macrocell (3GPP)
Model Arguments	Not Applicable
Shadow Fading	Suburban Macrocell (3GPP) Default

Attribute: RIP Parameters

Process Parameters	(...)
Interface Information	(...)
Aggregate Interfaces	None
Tunnel Interfaces	None
VLAN Interfaces	None
BVI Interfaces	None
Service Interfaces	None

Attribute: RIP Parameters.Process Parameters

Address Family	IPv4 - Any
Routing Instance / VRF Name	Global
Process Parameters	(...)

Attribute: RIP Parameters.Process Parameters.Process Parameters

Start Time	uniform (5, 10)
Timers	Default
Failure Impact	Retain Route Table
Network Information	None
Version	Version 1
Auto Summary	Enabled
Default Information Originate	Disabled
Send Style	Broadcast
Redistribution	Disabled
Route Filters	None
Routing Policies	None
Offset Lists	None
Administrative Weight	120
Administrative Weight (Prefix)	None
Process Tag	1

Attribute: RIP Parameters.Interface Information

Index	Name	Status	Silent Mode	Advertisement Mode	Cost	Send Version	Receive Version	Triggered Extension	Subinterface Information	Route Filters	Offset Lists	Process Tag(s)
0	IF0	Enabled	Disabled	Split Horizon with Poison Reverse	1	Default	Default	Disabled	None	None	None	1
1	IF1	Enabled	Disabled	Split Horizon with Poison Reverse	1	Default	Default	Disabled	None	None	None	1
2	IF2	Enabled	Disabled	Split Horizon with Poison Reverse	1	Default	Default	Disabled	None	None	None	1
3	IF3	Enabled	Disabled	Split Horizon with Poison Reverse	1	Default	Default	Disabled	None	None	None	1

Attribute: RIPng Parameters

Start Time	constant (5)
Stop Time	65.0
Timers	Default
Failure Impact	Retain Route Table
Interface Information	None
Tunnel Interfaces	None
VLAN Interfaces	None
Administrative Weight	120
Route Filters	None
Redistribution	Disabled

Attribute: System Information

Module Name	Unknown
Serial Number	Unknown
OS Type	Unknown
OS Version	Unknown
DRAM	Unknown
NVRAM	Unknown
Flash RAM	Unknown
Import Type	Config and MIB

VNES Access Address	Unknown

Attribute: TCP Parameters

Host Operating System	Windows 7
Flavor	New Reno
Maximum Segment Size	Auto-Assigned
Receive Buffer	Auto-Tuning
Receive Buffer Adjustment	Windows Based
Receive Buffer Usage Threshold	0.0
Delayed ACK Mechanism	Segment/Clock Based
Maximum ACK Delay	0.200
Maximum ACK Segments	2
Slow-Start Initial Count	2
Duplicate ACK Threshold	3
Window Scaling	Enabled
Selective ACK (SACK)	Enabled
Duplicate SACK (D-SACK)	Enabled
ECN Capability	Disabled
Segment Send Threshold	MSS Boundary
Active Connection Threshold	Unlimited
Nagle Algorithm	Enabled
Karn's Algorithm	Enabled
Timestamp	(...)
Initial Sequence Number	Auto Compute
Retransmission Thresholds	(...)
Initial RTO	1.0
Minimum RTO	0.25
Maximum RTO	30
RTT Gain	0.125
Deviation Gain	0.25
RTT Deviation Coefficient	4.0
Timer Granularity	0.5
Persistence Timeout	1.0
Connection Information	Do Not Print
Acceleration	Disabled

Attribute: TCP Parameters.Timestamp

Status	Passive
Clock Tick	500

Attribute: TCP Parameters.Retransmission Thresholds

Mode	Attempts Based
Maximum Connect Attempts	5
Maximum Data Attempts	6
Maximum Connect Interval	Not Applicable
Maximum Data Interval	Not Applicable

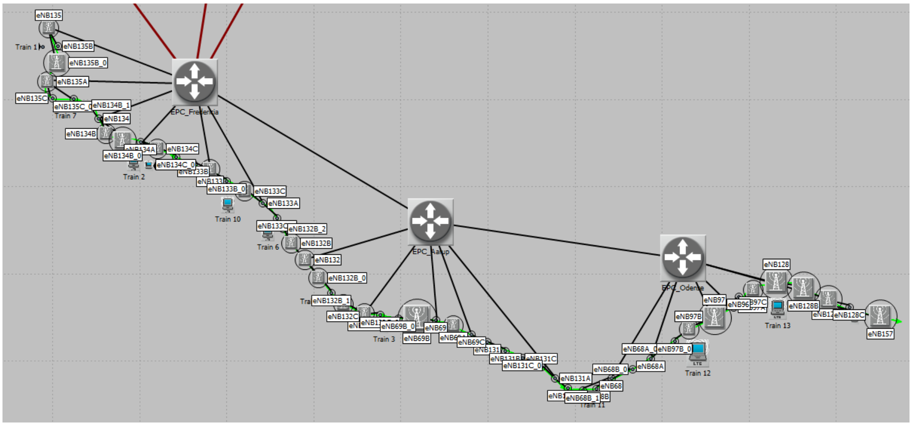


Figure B.3: The densest radio network deployment considered in the Snoghøj-Odense scenario. It consisted of 55 eNodeBs.

B.3 Details of the Copenhagen Central Station scenario

Table B.9: Uplink Jammer node attribute configuration. The node simulated uplink radio transmission in the neighbouring LTE cells.

Single Band Jammer

Full Name: Campus Network.JammerUplink01

name	JammerUplink01
Altitude	4.0
Jammer Band Base Frequency	0.878
Jammer Bandwidth	5.0
Jammer Bandwidth Usage Percentage	Full Bandwidth
Jammer Data Rate	14,000,000
Jammer Packet Interarrival Time	constant (0.0003)
Jammer Packet Size	exponential (5000)
Jammer Start Time	120
Jammer Stop Time	Infinity
Jammer Transmission Band Position	Bottom
Jammer Transmitter Power	4.0

Table B.10: Downlink Jammer node attribute configuration. The node simulated downlink radio transmission in the neighbouring LTE cells.

Single Band Jammer

Full Name: Campus Network.JammerDownlink01

name	JammerDownlink01
Altitude	50
Jammer Band Base Frequency	0.923
Jammer Bandwidth	5.0
Jammer Bandwidth Usage Percentage	Full Bandwidth
Jammer Data Rate	14,000,000
Jammer Packet Interarrival Time	constant (0.002)
Jammer Packet Size	exponential (250)
Jammer Start Time	120
Jammer Stop Time	Infinity
Jammer Transmission Band Position	Top
Jammer Transmitter Power	4.0

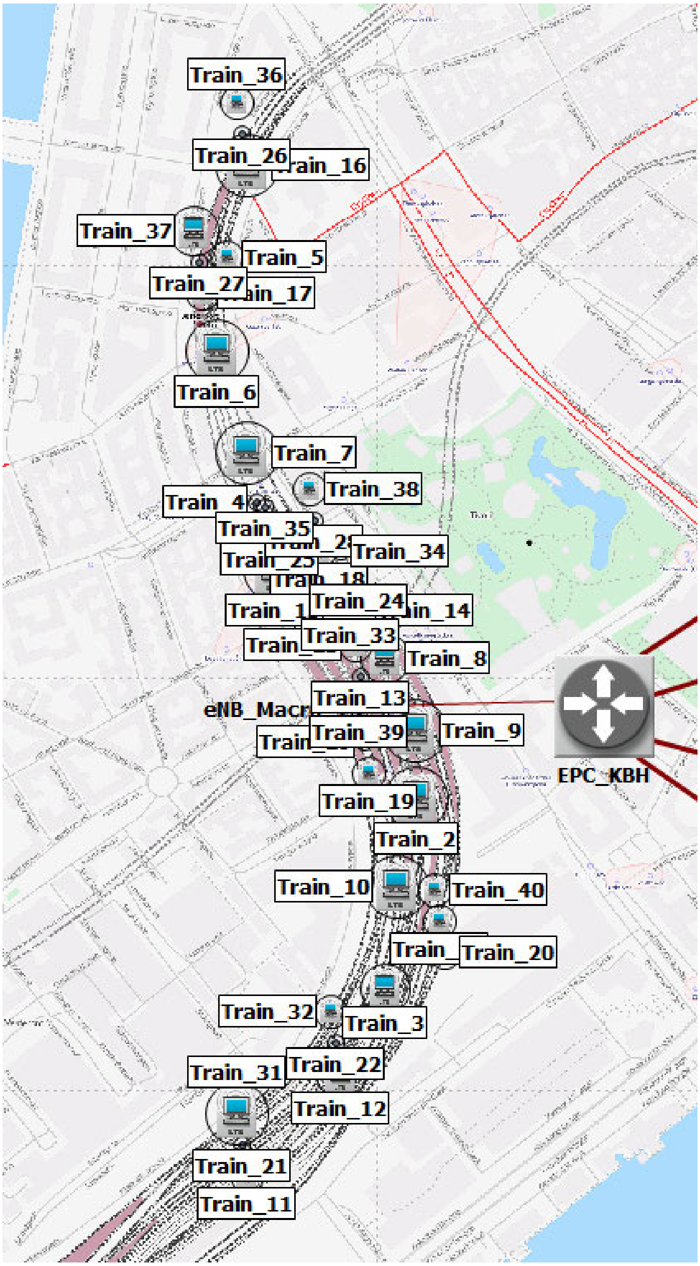


Figure B.4: UE (train) distribution at Copenhagen Central Station. The case with 40 UEs is shown.

Table B.11: eNodeB node attribute configuration used in Copenhagen Central Station simulation scenario.

LTE eNodeB

Full Name: Campus Network.eNB_Macro_1

name	eNB_Macro_1
AAA Parameters	Not Configured
APS Parameters	None
ARP Parameters (IF0 P0)	Default
ARP Parameters (IF1 P0)	Default
ARP Parameters (IF2 P0)	Default
ARP Parameters (IF3 P0)	Default
Address	Auto Assigned
Admission Control Parameters	Default
Antenna Gain (dBi)	15 dBi
BGP Based	None
BGP Parameters	Default
BGP Routing Table	Do Not Export
Battery Capacity	Unlimited
Buffer Status Report Parameters	Default
CPU Background Utilization	None
CPU Resource Parameters	Single Processor
CPU Utilization	Unassigned
CQI Transmission Parameters	Default
Cross Connect Groups	None
Cross-Connects Parameters	None
Customers	Not Configured
DHCPv6 Client Parameters	Disabled
DHCPv6 Server Parameters	Disabled
DLSw+ Parameters	None
DNS Parameters	None
DRX Parameters	Default
DVMRP Parameters	Not Configured
Delay, Jitter and Loss	Unassigned
EIGRP Parameters	(...)
EIGRP Routing Table	Do Not Export
EPCs Served	All
Ethernet Parameters (IF0 P0)	Default (Host)
Ethernet Parameters (IF1 P0)	Default (Host)
Ethernet Parameters (IF2 P0)	Default (Host)
Ethernet Parameters (IF3 P0)	Default (Host)
HSRP Operational Data	Unknown
HSRP Parameters	Not Configured
Handover Parameters	Default
IGMP Operational Data	None
IGMP Parameters	Not Configured
IGRP Parameters	(...)
IGRP Routing Table	Do Not Export
IP Forwarding Table	Do Not Export
IP Multicast Group-to-RP Table	Do Not Export
IP Multicast Parameters	Not Configured
IP QoS Parameters	None
IP Routing Parameters	(...)
IPSec Parameters	None
IPX Parameters	None
IPv6 Parameters	None
IS-IS Parameters	(...)

IS-IS Routing Table	Do Not Export
Kerberos Parameters	Not Configured
L1/L2 Control Parameters	Default
L2TP Control Channel Parameters	None
LACP System Priority	32768
LDP Based	None
LDP Parameters	(...)
Line Information	None
Logging	Disabled
MBSFN Area	Disabled
MPLS Parameters	(...)
MSDP Parameters	Not Configured
Maximum Transmission Power (W)	1.0
NAT Parameters	Not Configured
NHRP Parameters	None
Number of Receive Antennas	2
Number of Transmit Antennas	2
OSPF Link State Database	Do Not Export
OSPF Parameters	(...)
OSPF Routing Table	Do Not Export
OSPFv3 Parameters	(...)
Operating Power	20
Operational HTTP Parameters	Not Configured
PDCP Compression	Disabled
PHY Profile	LTE 5 MHz FDD 900 MHz
PIM Parameters	Not Configured
PIM-DVMRP Interoperability Parameters	Not Configured
PIM-SM Routing Table	Do Not Export
Pathloss Parameters	(...)
Pseudowire Classes	None
RADIUS Parameters	Not Configured
RAM Utilization	Unassigned
RIP Parameters	(...)
RIP Routing Table	Do Not Export
RIPng Parameters	(...)
RIPng Routing Table	Do Not Export
RSRB Parameters	None
RSVP Protocol Parameters	(...)
Random Access Parameters	Default
Receiver Sensitivity (dBm)	-200dBm
SNMP Parameters	Disabled
SRP Parameters	Not Configured
Scheduling Mode	Link Adaptation and Channel Dependent Scheduling
Service Distribution Points	Not Configured
Serving MBMS EPC ID	0
Subscriber Services	Not Configured
System Information	(...)
TACACS+ Parameters	Not Configured
TCP Parameters	Windows 7
Transparent Bridge Parameters	None
User Access Control	Not Configured
VRF Instances	None
VRF Table	Do Not Export
VRRP Parameters	Not Configured
WLAN Configuration	Not Configured
X2 Capability	Enabled

eNodeB ID	1
eNodeB Selection Threshold	-100
lte_as.Sector Number	No Sectors
lte_s1.Sector Number	No Sectors
sysmgt.Device Availability	Unassigned
sysmgt.Fabric Extenders	promoted
sysmgt.Interface Availability	Unnasigned
sysmgt.Packet Errors and Discards	Unassigned
sysmgt.Throughput	Unassigned
sysmgt.Traffic Statistics	Unassigned
sysmgt.VDC Configuration	Not Configured

Attribute: IP Routing Parameters

Router ID	Auto Assigned
Autonomous System Number	Auto Assigned
Interface Information	(...)
Aggregate Interfaces	None
Loopback Interfaces	(...)
Tunnel Interfaces	None
VLAN Interfaces	None
BVI Interfaces	None
Service Interfaces	None
IPSec Connections	None
Controller Configuration	None
Default Gateway	Unassigned
Default Network(s)	None
Static Routing Table	None
Static Routes Across VRFs	Enabled
Load Balancing Options	Destination-Based
Multipath Routes Threshold	Unlimited
Administrative Weights	(....)
OS Version	Not Set
Standard ACL Configuration	None
Extended ACL Configuration	None
Damping Configuration	None
AS Path Lists	None
Community Lists	None
Extended Community Lists	None
Prefix Filter Configuration	None
Route Map Configuration	None
Route Policy Configuration	None
Firewall Filter Configuration	None
Local Policy	None
Forwarding Table Policies	Not Configured
Fate Sharing	Not Configured
IPv4 Configurations	Default
IP Domain Lookup	Not Configured
Tunnel Policy Configuration	Not Configured

Attribute: IP Routing Parameters.Loopback Interfaces

Name	LB0
Status	Active
Operational Status	Infer
Address	Auto Assigned
Subnet Mask	Auto Assigned

Secondary Address Information	Not Used
Routing Protocol(s)	RIP
MTU	Ethernet
MTU Intended	disabled
Protocol MTUs	(...)
Metric Information	Default
Packet Filter	None
Policy Routing	None
Routing Instance	None
VRF Sitemap	None
Interface Configurations	Default
Description	N/A

Attribute: OSPF Parameters

Processes	Default
Interface Information	(...)
Aggregate Interfaces	None
Loopback Interfaces	(...)
Tunnel Interfaces	None
VLAN Interfaces	None
BVI Interfaces	None

Attribute: OSPF Parameters.Interface Information.Timers [0]

Hello Interval	10
Router Dead Interval	40
Interface Transmission Delay	1.0
Retransmission Interval	5.0

Attribute: OSPF Parameters.Interface Information.MANET Parameters [0]

Full Hello Frequency	Every Third
Hello Repeat Count	Thrice
Adjacency Connectivity	Biconnected
MDR Constraint	3

Attribute: OSPF Parameters.Interface Information.Timers [1]

Hello Interval	10
Router Dead Interval	40
Interface Transmission Delay	1.0
Retransmission Interval	5.0

Attribute: OSPF Parameters.Interface Information.MANET Parameters [1]

Full Hello Frequency	Every Third
Hello Repeat Count	Thrice
Adjacency Connectivity	Biconnected
MDR Constraint	3

Attribute: OSPF Parameters.Interface Information.Timers [2]

Hello Interval	10
Router Dead Interval	40
Interface Transmission Delay	1.0
Retransmission Interval	5.0

Attribute: OSPF Parameters.Interface Information.MANET Parameters [2]

Full Hello Frequency	Every Third
Hello Repeat Count	Thrice
Adjacency Connectivity	Biconnected
MDR Constraint	3

Attribute: OSPF Parameters.Interface Information.Timers [3]

Hello Interval	10
Router Dead Interval	40
Interface Transmission Delay	1.0
Retransmission Interval	5.0

Attribute: OSPF Parameters.Interface Information.MANET Parameters [3]

Full Hello Frequency	Every Third
Hello Repeat Count	Thrice
Adjacency Connectivity	Biconnected
MDR Constraint	3

Attribute: OSPF Parameters.Loopback Interfaces

Name	LB0
Status	Disabled
Silent Mode	Disabled
Area ID	Specify ...
Process Tag(s)	1

Attribute: OSPFv3 Parameters

Processes	Default
Interface Information	(...)
Aggregate Interfaces	None
Loopback Interfaces	None
Tunnel Interfaces	None
VLAN Interfaces	None

Attribute: Pathloss Parameters

Pathloss Model	Urban Macrocell (3GPP)
Model Arguments	Not Applicable
Shadow Fading	Urban Macrocell (3GPP) Default

Attribute: RIP Parameters

Process Parameters	(...)
Interface Information	(...)
Aggregate Interfaces	None
Tunnel Interfaces	None
VLAN Interfaces	None
BVI Interfaces	None
Service Interfaces	None

Attribute: RIP Parameters.Process Parameters

Address Family	IPv4 - Any
Routing Instance / VRF Name	Global
Process Parameters	(...)

Attribute: RIP Parameters.Process Parameters.Process Parameters

Start Time	uniform (5, 10)
Timers	Default
Failure Impact	Retain Route Table
Network Information	None
Version	Version 1
Auto Summary	Enabled
Default Information Originate	Disabled
Send Style	Broadcast
Redistribution	Disabled
Route Filters	None
Routing Policies	None
Offset Lists	None
Administrative Weight	120
Administrative Weight (Prefix)	None
Process Tag	1

Attribute: RIP Parameters.Interface Information

Index	Name	Status	Silent Mode	Advertisement Mode	Cost	Send Version	Receive Version	Triggered Extension	Subinterface Information	Route Filters	Offset Lists	Process Tag(s)
0	IF0	Enabled	Disabled	Split Horizon with Poison Reverse	1	Default	Default	Disabled	None	None	None	1
1	IF1	Enabled	Disabled	Split Horizon with Poison Reverse	1	Default	Default	Disabled	None	None	None	1
2	IF2	Enabled	Disabled	Split Horizon with Poison Reverse	1	Default	Default	Disabled	None	None	None	1
3	IF3	Enabled	Disabled	Split Horizon with Poison Reverse	1	Default	Default	Disabled	None	None	None	1

Attribute: RIPng Parameters

Start Time	constant (5)
Stop Time	65.0
Timers	Default
Failure Impact	Retain Route Table
Interface Information	None
Tunnel Interfaces	None
VLAN Interfaces	None
Administrative Weight	120
Route Filters	None
Redistribution	Disabled

Attribute: RSVP Protocol Parameters

Waiting Time	1.0
Refresh Interval	30
Lifetime Multiplier	3
Blockade Multiplier	1.0
Preemption	Normal
Authentication	Disabled
Neighbor Configuration	Not Configured
Graceful Restart	Disabled

Prefix Filtering	Not Configured
Interface Information	(...)
Aggregate Interfaces	None
Tunnel Interfaces	None
VLAN Interfaces	None
BVI Interfaces	None

Attribute: System Information

Module Name	Unknown
Serial Number	Unknown
OS Type	Unknown
OS Version	Unknown
DRAM	Unknown
NVRAM	Unknown
Flash RAM	Unknown
Import Type	Config and MIB
VNES Access Address	Unknown

Bibliography

[1] G. Theeg and S. Vlasenko, eds., *Railway Signalling and Interlocking*. Germany: Eurailpress, first ed., 2009. ISBN 978-3-7771-0394-5.

[2] "The Electro Magnetic Telegraph of the Great Western Railway," in *Journal of the Franklin Institute*, vol. 29(4), pp. 271–272, 1840.

[3] R. Stephenson, "On the Railways and Telegraphs of Great Britain," in *Journal of the Franklin Institute*, vol. 61(4), pp. 217–221, 1856.

[4] P. Stanley, ed., *ETCS for Engineers*. Germany: Eurailpress, first ed., 2011. ISBN 978-3-7771-0416-4.

[5] R. Pascoe and T. Eichorn, "What is Communication-Based Train Control?," *IEEE Vehicular Technology Magazine*, vol. 4(4), pp. 16–21, December 2009.

[6] M. Fitzmaurice, "Wayside Communications: CBTC Data Communications Subsystems," *IEEE Vehicular Technology Magazine*, vol. 8(3), pp. 73–80, September 2013.

[7] A. Grogan, "Driverless trains: It's the automatic choice!," *IET Engineering & Technology*, vol. 7(5), pp. 54–57, June 2012.

[8] M. Aguado, O. Onandi, P. S. Agustin, M. Higuero, and E. J. Taquet, "WiMAX on Rails," *IEEE Vehicular Technology Magazine*, vol. 3(3), pp. 47–56, September 2008.

[9] N. Yanase, "Necessities for future high speed rolling stock," tech. rep., International Union of Railways (UIC), Paris, France, January 2010. ISBN 978-2-7461-1931-4.

[10] J. Calle-Sanchez, M. Molina-Garcia, J. I. Alonso, and A. Fernandez-Duran, "Long Term Evolution in High Speed Railway Environments: Feasibility and Challenges," *Bell Labs Technical Journal, Special Issue: Next-Generation Wireless Technologies*, vol. 18(2), pp. 237–253, September 2013.

[11] D. Taylor, N. Lofmark, and M. McKavanagh, "Survey on operational communications (study for the evolution of the railway communications system)," Final report for the European Railway Agency 37760-496v04, Analysys Mason, February 2014. Available online: http://www.era.europa.eu/Document-Register/Pages/Analysys-Mason-final-report-for-ERA.aspx.

[12] P. Winter, ed., *Compendium on ERTMS*. Eurailpress, first ed., 2009. ISBN 978-3-7771-0396-9.

[13] S. F. Ruesche, J. Steuer, and K. Jobmann, "The European Switch. A Packet-Switched Approach to a Train Control System," *IEEE Vehicular Technology Magazine*, vol. 3(3), pp. 37–46, September 2008.

[14] K. Kastell, S. Bug, A. Nazarov, and R. Jakoby, "Improvments in Railway Communication via GSM-R," in *Proceedings of the 63rd IEEE Vehicular Technology Conference (VTC 2006-Spring)*, pp. 3026–3030, IEEE, 2006.

[15] European Council, "Council Directive 96/48/EC of 23 July 1996 on the interoperability of the trans-European high-speed rail system," *Official Journal of the European Communities*, no. L 235, pp. 6–24, 1996.

[16] European Commission, "Commission Decision 2001/260/EC of 21 March 2001 on the basic parameters of the command-control and signalling subsystem of the trans-European high-speed rail system referred to as 'ERTMS characteristics' in Annex II(3) to Directive 96/48/EC," *Official Journal of the European Union*, no. L 93, pp. 53–56, 2001.

[17] V. Carpinelli, A. Missoumi, E. Brutin, and C. Filippini, *ERTMS Atlas 2012*. International Union of Railways (UIC), third ed., 2012. ISBN 978-2-7461-2078-5.

[18] E. Brutin, "ERTMS: Global dimensions, global challenges," *European Railway Review*, vol. 18(3), pp. 36–38, 2012.

[19] D. G. Fisher, "Requirements on the GSM-R Network for ETCS Support. Capacity, Performance and RAM," Ref: FSI85-222-007, Banedanmark, Rambøll, April 2008.

[20] The Signalling Programme, "Signalling System Model. F-bane," Ref: FSI85-222-007, Banedanmark, Rambøll, April 2008.

[21] H. Hofestadt, "GSM-R: Global System for Mobile radio communications for Railways," in *Proceedings of the International Conference on Electric Railways in a United Europe*, pp. 111–115, IET, March 1995.

[22] Project EIRENE, "System Requirements Specification," Ref: P0011D010, v. 15.1, International Union of Railways (UIC), June 2010. ISBN 2-7461-1832-4.

[23] Project EIRENE, "Functional Requirements Specification," Ref: P0011D009, v. 7.1, International Union of Railways (UIC), June 2010. ISBN 2-7461-1831-7.

[24] I. Ljubic and D. Simunic, "Advanced Speech Call Items for GSM-Railway," in *Proceedings of the 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE)*, (Aalborg, Denmark), pp. 131–136, IEEE, 2009.

[25] M. Sauter, *From GSM to LTE: an introduction to mobile networks and mobile broadband*. John Wiley & Sons, first ed., 2011. ISBN 978-0470667118.

[26] A. Neele and S. Wootton, "GSM-R Procurement Guide," Tech. Rep. CLA111D005-5.0, International Union of Railways (UIC), Paris, France, February 2007. ISBN 2-7461-1315-5.

[27] C. Hamling, "An overview of GSM data services," in *IEE Colloquium: Roaming with Data*, (London), pp. 2/1–2/8, IET, March 1995.

[28] S. Åkesson, "GPRS, General Packet Radio Service," in *Proceedings of the Fourth IEEE International Conference on Universal Personal Communications*, (Tokyo, Japan), pp. 640–643, IEEE, November 1995.

[29] R. Ludwig and B. Rathonyi, "Link Layer Enhancements for TCP/IP over GSM," in *Proceedings of INFOCOM'99: Conference on Computer Communications*, (New York, USA), pp. 415–422, IEEE, 1999.

[30] Union Industry of Signalling (UNISIG), "ERTMS/ETCS. SUBSET-093. GSM-R Interfaces. Class 1 requirements," Tech. Rep. v. 2.3.0, October 2005.

[31] Radio Spectrum Committee, "Spectrum Requirements for Railway Applications," Working Document RSCOM10-45, European Commission - Information Society and Media Directorate-General, Brussels, Belgium, September 2010.

[32] R. Qureshi, "Ericsson Mobility Report on the Pulse of the Networked Society," Tech. Rep. EAB-14:028658 Uen, Revision A, Ericsson AB, Stockholm, Sweden, June 2014. Available online: http://www.ericsson.com/mobility-report.

[33] F. Pujol and J. S. Marcus, "Evolution of GSM-R. ERA/2014/04/ERTMS/OP," Interim Report v 1.5, IDATE, DigiWorld Institute, WIK Consult, January 2015. Available online: http://www.era.europa.eu/Document-Register/Documents/ERA%20Evolution%20of%20GSMR%20Interim%20report%20V1%205.pdf.

[34] European Railway Agency (ERA), *ERA Program for the evolution of the radio communication system for railways*. May 2014. Available online: http://www.era.europa.eu/Document-Register/Pages/ERA-Program-for-the-evolution-of-the-radio-communicationsystem-for-railways.aspx.

[35] Teltronic, "Kazakhstan National Railway selects TELTRONIC to implement the world's first ETCS-based rail signaling system over TETRA." Available online: http://www.teltronic.es/en/node/377, March 2012. Press release.

[36] J. Schiller, *Mobile Communications*. Addison-Wesley, second ed., 2003.

[37] "WiMAX Advanced: Deployment Scenarios Based on Input from WiMAX Operators and Vendors," white paper, WiMAX Forum, September 2014.

[38] S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution From Theory to Practice*. John Wiley & Sons, 2011. ISBN 978-0-470-66025-6.

[39] H. Holma and A. Toskala, *LTE for UMTS - OFDMA and SC-FDMA Based Radio Access*. John Wiley & Sons, first ed., 2009. ISBN 978-0-470-99401-6.

[40] J. Calle-Sanchez, M. Molina-Garcia, and J. I. Alonso, "Top challenges of LTE to become the next generation railway communication system," in *Proceedings of the Thirteenth International Conference on Design and Operation in Railway Engineering (COMPRAIL 2012)*, (New Forest, UK), pp. 85–96, WIT Press, September 2012.

[41] M. Liem and V. B. Mendiratta, "Mission Critical Communication Networks for Railways," *Bell Labs Technical Journal*, vol. 13(3), pp. 29–46, 2011.

[42] A. D. Zayas, C. A. G. Perez, and P. M. Gomez, "Third-Generation Partnership Project Standards: For Delivery of Critical Communications for Railways," *IEEE Vehicular Technology Magazine*, vol. 9(2), pp. 58–68, June 2014.

[43] Alcatel-Lucent, "Changing track: ground-to-train broadband communications and new train-borne applications are considering a move to 4G/LTE," *Railway Strategies*, pp. 6–7, November 2009.

[44] G. Barbu, "E-Train, Broadband Communications with Moving Trains," tech. rep., International Union of Railways (UIC), Paris, France, June 2010. ISBN 978-2-7461-1849-2.

[45] K. Guan, Z. Zhong, and B. Ai, "Assessment of LTE-R using High Speed Railway Channel Model," in *Proceedings of the 3rd International Conference on Communications and Mobile Computing (CMC)*, pp. 461–464, IEEE, 2011. ISBN 978-1-61284-312-4.

[46] A. Parichehreh and U. Spagnolini, "Seamless LTE connectivity in High Speed Trains," in *Proceedings of Wireless Communications and Networking Conference (WCNC)*, (Istanbul, Turkey), pp. 2067–2072, IEEE, 2014.

[47] W. Luo, X. Fang, M. Cheng, and X. Zhou, "An optimized handover trigger scheme in LTE systems for high-speed railway," in *Proceedings of the Fifth International Workshop on Signal Design and its Applications in Communications (IWSDA)*, (Guilin, China), pp. 193–196, IEEE, 2011. ISBN 978-1-61284-047-5.

[48] Y. Feng, R. Xu, and Z. Zhong, "Channel Estimation and ICI Cancellation for LTE Downlink in High-speed Railway Environment," in *Proccedings of the 11th International Conference on Signal Processing (ICSP)*, vol. 2, (Beijing, China), pp. 1419–1423, IEEE, 2012. ISBN 978-1-4673-2196-9.

[49] G. Wu, H. Yuan, W. Guo, S. Hu, and S. Li, "On design of physical random access channel in high-speed railway LTE systems," *Chinese Science Bulletin*, vol. 59(35), pp. 5042–5050, October 2014.

[50] G. Tomasov, M. Wu, J. Wen, and H. Liu, "LTE fixed-point handover algorithm for high-speed railway scenario," in *Proceedings of the 3rd International Conference on Computer Science and Network Technology (ICCSNT)*, pp. 919–923, IEEE, 2013.

[51] N. Gresset, J. Letessier, and H. Bonneville, "A QoS-Based Multi-user Scheduler Applied to Railway Radio-Communications," in *Proceedings of the 5th International Workshop on Communication Technologies for Vehicles, Nets4Cars/Nets4Trains 2013*, vol. 7865 of *Lecture Notes in Computer Science*, (Villeneuve d'Ascq, France), pp. 31–45, Springer, May 2013. ISBN 978-3-642-37973-4.

[52] A. Khayat, M. Kassab, M. Berbineau, M. A. Abid, and A. Belghith, "LTE Based Communication System for Urban Guided-Transport: A QoS Performance Study," in *Proceedings of the 5th International Workshop on Communication Technologies for Vehicles, Nets4Cars/Nets4Trains 2013*, vol. 7865 of *Lecture Notes in Computer Science*, (Villeneuve d'Ascq, France), pp. 197–210, Springer, May 2013. ISBN 978-3-642-37973-4.

[53] L. Hansen, P. Booth, and W. Lautenschlager, "ETCS/GSM-R Quality of Service - Operational Analysis," Tech. Rep. EEIG: 04E117, EEIG ERTMS Users Group, October 2005.

[54] J. Yuan and I. A. Hansen, "Optimizing capacity utilization of stations by estimating knock-on train delays," *Transportation Research Part B: Methodological*, vol. 41(2), pp. 202–217, 2007. Advanced Modelling of Train Operations in Stations and Networks.

[55] European Railway Agency (ERA), "ERTMS/ETCS. Radio Transmission FFFIS for EuroRadio," Form Fit Functional Interface Specification A 11 T 6001 v. 12.4, 2012. Originated by EuroRadio Interface Group.

[56] I. Lopez, M. Aguado, and E. Jacob, "End-to-End Multipath Technology: Enhancing Availability and Reliability in Next-Generation Packet-Switched Train Signaling Systems," *IEEE Vehicular Technology Magazine*, vol. 9(1), pp. 28–35, March 2014.

[57] Trafikverket, "Coexistence between GSM-R and 3G/4G-Systems in the 900 MHz Frequency Band - Swedish View," Tech. Rep. TRV 2013/13976, Sweden, March 2013.

[58] J. Binningsbø, R. Baldersheim, and J. P. Lauvstad, "GSM-R Radio Planning Guidelines," Tech. Rep. 3A-GSM-038, Jernbaneverket Utbygging, Norway, February 2006.

[59] Riverbed Technology, *OPNET Modeler, v 17.5.A PL5*. 2013.

[60] 3GPP, "Spatial channel model for Multiple Input Multiple Output (MIMO) simulations," TS 25.996, v 8.0.0, 2008.

[61] Trafikministeriet, "Modernisering af jernbanens hovednet," technical report, April 1997. (In Danish). Available online: http://www.trm.dk/da/publikationer/1997/modernisering-af-jernbanens-hovednet.

[62] Erhvervsstyrelsen (the Danish Business Authority), *Mastedatabasen*. Available online: www.mastedatabasen.dk.

[63] COWI, "VVM-screening af radiokommunikationsmaster. Oversigtskort. Placering af GSM-R master," Tech. Rep. 72648A, Banedanmark, September 2010.

[64] Banedanmark, "Banedanmark udskifter radiomast ved Fruens Bøge," press release, May 2011. (In Danish). Available online: www.bane.dk/hentmedie.asp?filID=9563.

[65] B. Dusza, C. Ide, P.-B. Bok, and C. Wietfeld, "Optimized cross-layer protocol choices for LTE in high-speed vehicular environments," in *Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC 2013)*, pp. 1046–1051, IEEE, 2013. ISBN 978-1-4673-2480-9.

[66] J. Salo, M. Nur-Alam, and K. Chang, "Practical Introduction to LTE Radio Planning," June 2013. Available online: http://www.lteexpert.com/lte_rf_wp_16June2013.pdf.

[67] 3GPP, "Requirements for Evolved UTRA (E-UTRA) and Evolved UTRAN (E-UTRAN)," TS 25.913, v 8.0.0, 2008.

[68] F. J. Martín-Vega, I. M. Delgado-Luque, F. Blánquez-Casado, G. Gómez, M. C. Aguayo-Torres, and J. T. Entrambasaguas, "LTE Performance over High Speed Railway Channel," in *Proceedings of the 78th Vehicular Technology Conference (VTC Fall)*, pp. 1–5, IEEE, September 2013.

[69] K. Dimou, M. Wang, Y. Yang, M. Kazmi, A. Larmo, J. Pettersson, W. Muller, and Y. Timner, "Handover within 3GPP LTE: Design Principles and Performance," in *Proceedings of the 70th Vehicular Technology Conference Fall (VTC 2009-Fall)*, IEEE, September 2009. ISBN 978-1-4244-2514-3.

[70] J. Li, L. Tian, Y. Zhou, and J. Shi, "An Adaptive Handover Trigger Scheme for Wireless Communications on High Speed Rail," in *Proceedings of International Conference on Communications (ICC 2012)*, pp. 5185–5189, IEEE, 2012. ISBN 978-1-4577-2052-9.

[71] U. Æ. Christiansen, "Stationskapaciteten ved København H," tech. rep., Trafikstyrelsen, 2013. (In Danish).

[72] H. E. Jørgensen, "Kapacitetsforbedringsplan. Overbelastet infrastruktur," tech. rep., Banedanmark, Denmark, January 2011. (In Danish). Available online: http://www.bane.dk/visSideforloebBeholder.asp?artikelID=13401.

[73] D. T. Fokum and V. S. Frost, "A Survey on Methods for Broadband Internet Access on Trains," *IEEE Communications Surveys & Tutorials*, vol. 12(2), pp. 171–185, 2010.

[74] M. R. Nielsen, "Besvælige iPads forsinkede S-tog," *Jyllands-Posten*, November 27, 2014. (In Danish). Available online: http://jyllands-posten.dk/indland/trafik/ECE7239850/Besv%C3%A6rlige-iPads-forsinkede-S-tog/.

[75] 3GPP, "Policy and charging control architecture (Release 8)," TS 23.203, v 8.9.0, 2010.

[76] A. Larmo, M. Lindström, M. Meyer, G. Pelletier, J. Torsner, and H. Wiemann, "The LTE Link-Layer Design," *IEEE Communications Magazine*, vol. 47(4), pp. 52–59, April 2009.

[77] A. Asheralieva, K. Mahata, and J. Y. Khan, "Delay and Loss Due to Uplink Packet Scheduling in LTE Network," in *Wireless Access Flexibility. Proceeedings*

of the First International Workshop (WiFlex 2013), vol. 8072 of *Lecture Notes in Computer Science*, (Kaliningrad, Russia), pp. 1–12, Springer, September 2013.

[78] E. D. Signore, R. Giuliano, F. Mazzenga, M. Petracca, M. Vari, F. Vatalaro, A. Neri, and F. Rispoli, "On the Suitability of Public Mobile Networks for Supporting Train Control/Management Systems," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC 2014)*, pp. 3302–3307, IEEE, 2014.

[79] D. Lowther, "GSMA Leads Mobile Industry Towards a Single, Global Solution for Voice over LTE," press release, GSM Association (GSMA), February 15, 2010. Available online: http://www.gsma.com/technicalprojects/gsma-leads-mobile-industry-towards-a-single-global-solution-for-voice-over-lte.

[80] GSM Association (GSMA), "IMS Profile for Voice and SMS," Official Document IR.92 v 7.0, 2013.

[81] 3GPP, "IP Multimedia Subsystem (IMS); Stage 2 (Release 13)," TS 23.228, v 13.0.0, 2014.

[82] M. Anehill, M. Larsson, G. Strömberg, and E. Parsons, "Validating Voice over LTE end-to-end," *Ericsson Review*, vol. 1, 2012.

[83] ITU Telecommunication Standardization Sector (ITU-T), *Series G: Transmission Systems And Media, Digital Systems And Networks. Recommendation G.114. International telephone connections and circuits - General Recommendations on the transmission quality for an entire international telephone connection. One-way transmission time.* International Telecommunication Union (ITU), May 2003.

[84] A. Teković, I. Pešut, and Z. Morić, "Voice Service in an LTE Network - CSFB," in *Proceedings of the 55th International Symposium ELMAR-2013*, (Croatia), pp. 251–254, IEEE, 2013.

[85] M. Poikselkä, H. Holma, J. Hongisto, J. Kallio, and A. Toskala, *Voice over LTE (VoLTE)*. John Wiley & Sons, 2012. ISBN 978-1-119-95168-1.

[86] 3GPP, "IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3 (Release 12)," TS 24.229, v 12.6.0, 2014.

[87] G. Camarillo and M.-A. Garcia-Martin, *The 3G IP Multimedia Subsystem (IMS). Merging the Internet and the Cellular Worlds*. John Wiley & Sons, September 2008. ISBN 978-0-470-51662-1.

[88] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” Request for Comments 3550, July 2003.

[89] ITU Telecommunication Standardization Sector (ITU-T), *Series P: Telephone Transmission Quality. Recommendation P.800. Methods For Subjective Determination Of Transmission Quality*. International Telecommunication Union (ITU), August 1996.

[90] ITU Telecommunication Standardization Sector (ITU-T), *Series G: Transmission Systems And Media, Digital Systems And Networks. Recommendation G.107. The E-model: a computational model for use in transmission planning*. International Telecommunication Union (ITU), December 2011.

[91] M. Heddebaut, S. Mili, D. Sodoyer, E. Jacob, M. Aguado, C. P. Zamalloa, I. Lopez, and V. Deniau, “Towards a resilient railway communication network against electromagnetic attacks,” in *Proceedings of the Transport Research Arena (TRA) 5th Conference: Transport Solutions from Research to Deployment*, (Paris, France), September 2014.

[92] European Commission, “Commission Decision 2008/671/EC of 5 August 2008 on the harmonised use of radio spectrum in the 5 875-5 905 MHz frequency band for safety-related applications of Intelligent Transport Systems (ITS),” *Official Journal of the European Union*, no. L 220, pp. 24–26, 2008.

[93] ITU Radiocommunication Sector (ITU-R), *M Series: Report M.2135-1 Guidelines for evaluation of radio interface technologies for IMT-Advanced*. International Telecommunication Union (ITU), December 2009.

[94] 3GPP, “Services and service capabilities,” TS 22.105, v 12.1.0, 2014.

[95] Y.-C. Chen, E. M. Nahum, R. J. Gibbens, D. Towsley, and Y. sup Lim, “Characterizing 4G and 3G Networks: Supporting Mobility with Multi-Path TCP,” Technical Report UM-CS-2012-022, University of Massachusetts Amherst (UMass Amherst), 2012. Available online: http://people.cs.umass.edu/~yungchih/.

Aleksander Sniady was born in Lodz, Poland, in 1987. He received his double M.Sc. degree in Telecommunications and Computer Science from Technical University of Denmark (DTU) and Lodz University of Technology (TUL), in 2011.

In 2012, he started his Ph.D. project at DTU Fotonik with the Networks Technology & Service Platforms research group. His work concerned future railway mobile communication technologies, which can become an alternative to GSM-Railways (GSM-R) in the next generation of the European Rail Traffic Management System (ERTMS).

DTU Fotonik
Networks Technology & Service Platforms
Technical University of Denmark

Ørsteds Plads 343
2800 Kgs. Lyngby
Denmark

Tel. +45 4525 6352
Fax +45 4593 6581

www.fotonik.dtu.dk

DOI: 10.11581/DTU:00000010